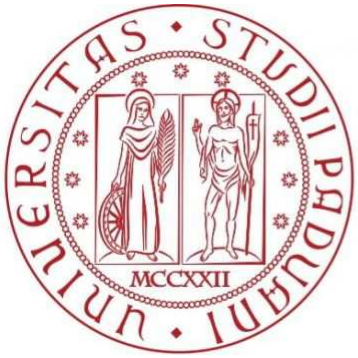

UNIVERSITÀ DEGLI STUDI DI PADOVA

**Facoltà di Ingegneria**

**Laurea magistreale Ingegneria dell'energia Elettrica**

# *Advanced techniques for rated speed increment in industrial drives*


**Relatore: Ch.mo Prof. Silverio Bolognani**

**Correlatore: Prof. Christopher Gerada**

**Laureando: Francesco Toso** **Matricola n° 1106839**


**Anno Accademico: 2015/2016**

The University of Nottingham

*…A Michelle, Vanna, Luciano, Filippo, Jacopo, Maria e Marinella….*

# Index

# INTRODUCTION

During this training and orientation project, I have taken part in the development of an industrial project called FACRI. My work was to write an algorithm of control able to satisfy precise specifics for a Permanent Magnet (PM) high speed motor, known the inverter and the motor parameters. Before to enter in details, it is useful doing a short introduction of the operating principles, the equations of a PM motor and the base structure of an actuator in order to have more clear ideas later on how the algorithm works.

***Operating principle***

The operating principle is based on the interactions between magnetic fields and eddy currents on the rotor and/or the stator. In particular PM motors have in the rotor permanent magnets characterized by a wide cycle of hysteresis and high residual magnetization that make magnetic flux in the air gap. The stator is built in the same way of an induction motor, with the conductors inside the cavities that when the current flows, they produce a stator magnetic field that summed with the PM one, makes a torque and therefore the rotation of the rotor. The rotor where there are the PM can be isotropic, so the reluctance crossing any directions is equal, and it's the case of a SPM (Surface Permanent Magnet) motor because the magnetic permeability of the PM is the same of the air, or it can present anisotropy that is the case of the IPM ( internal Permanent Magnet). In the next chapter will be described in detail the FACRI motor.

***Equations for a SPM motor***

To understand better let's analyse the motor equations neglecting the eddy currents and iron losses. The voltages of the phases a,b,c are the followings:

$$u_a = Ri_a + \frac{d\lambda_a}{dt} \qquad (0.1)$$

$$u_b = Ri_b + \frac{d\lambda_b}{dt} \qquad (0.2)$$

$$u_c = Ri_c + \frac{d\lambda_c}{dt} \qquad (0.3)$$

Where *R* is the resistance, *i* is the current and λ s the magnetic flux.

Applying the Clarke Transform we have:

$$\boldsymbol{u}_{\alpha\beta} = R\boldsymbol{i}_{\alpha\beta} + \frac{d\boldsymbol{\lambda}_{\alpha\beta}}{dt} \qquad (0.4)$$

Whose components are:

$$u_\alpha = Ri_\alpha + \frac{d\lambda_\alpha}{dt} \qquad (0.5)$$

$$u_\beta = Ri_\beta + \frac{d\lambda_\beta}{dt} \qquad (0.6)$$

The flux is given by two contributes, one by the PM and one by the stator currents, so if we apply the superposition principle we obtain:

$$\lambda_{mg_a} = \Lambda_{mg}\cos(\theta_{me}) \qquad (0.7)$$

$$\lambda_{mg_b} = \Lambda_{mg}\cos\left(\theta_{me} - \frac{2\pi}{3}\right) \qquad (0.8)$$

$$\lambda_{mg_c} = \Lambda_{mg}\cos\left(\theta_{me} - \frac{4\pi}{3}\right) \qquad (0.9)$$

Where

$$\theta_{me} = p\theta_m \qquad (0.10)$$

is the electric position of the rotor.

Applying the Park Transform:

$$\boldsymbol{\lambda}_{mg\ \alpha\beta} = \Lambda_{mg}e^{j\theta_{me}} \qquad (0.11)$$

Now using a rotating reference system the last space vector becomes:

$$\boldsymbol{\lambda}_{mg\ dq} = \boldsymbol{\lambda}_{mg\ \alpha\beta}e^{-j\theta_{me}} = \Lambda_{mg} \qquad (0.12)$$

The contribution given by the stator currents is the following:

$$\lambda_{a,i} = L_a i_a + L_{Mab} i_b + L_{Mac} i_c \qquad (0.13)$$

$$\lambda_{b,i} = L_{Mba} i_a + L_b i_b + L_{Mbc} i_c \qquad (0.14)$$

$$\lambda_{c,i} = L_{Mca} i_a + L_{Mcb} i_b + L_c i_c \qquad (0.15)$$

But with the hypothesis done before, we can assume that the auto inductances of the three phases are:

$$L_a = l_b = L_c = L_s \qquad (0.16)$$

While the mutual inductances are:

$$L_{Mab} = L_{Mac} = L_{Mba} = L_{Mbc} = L_{Mca} = L_{Mcb} = L_{Mss} = -|L_{Mss}| \qquad (0.17)$$

So we can write the phase a like follows:

$$\lambda_{a,i} = L_s i_a + L_{Mss}(i_b + i_c) \qquad (0.18)$$

But

$$i_b + i_c = -i_a \qquad (0.19)$$

And substituting

$$\lambda_{a,i} = (L_s - L_{Mss})i_a = Li_a \qquad (0.20)$$

$$\lambda_{b,i} = Li_b \qquad (0.21)$$

$$\lambda_{c,i} = Li_c \qquad (0.22)$$

Applying the Park Transform in a rotating reference system we have:

$$\boldsymbol{\lambda}_{i\,dq} = L\boldsymbol{i}_{dq} \qquad (0.23)$$

So the total flux in d-q is:

$$\boldsymbol{\lambda}_{dq} = \boldsymbol{\lambda}_{mg\,dq} + \boldsymbol{\lambda}_{i\,dq} = \Lambda_{mg} + L\boldsymbol{i}_{dq} \qquad (0.24)$$

Now skipping some calculations, we can write the final equations in abc:

$$u_a = Ri_a + L\frac{di_a}{dt} + \Lambda_{mg}\omega_{me}\cos\left(\theta_{me} + \frac{\pi}{2}\right) \qquad (0.25)$$

$$u_b = Ri_b + L\frac{di_b}{dt} + \Lambda_{mg}\omega_{me}\cos\left(\theta_{me} + \frac{\pi}{2} - \frac{2\pi}{3}\right) \qquad (0.26)$$

$$u_c = Ri_c + L\frac{di_c}{dt} + \Lambda_{mg}\omega_{me}\cos\left(\theta_{me} + \frac{\pi}{2} - \frac{4\pi}{3}\right) \qquad (0.27)$$

Applying the space vector definition we finally obtain:

$$\boldsymbol{u}_{\alpha\beta} = R\boldsymbol{i}_{\alpha\beta} + L\frac{d\boldsymbol{i}_{\alpha\beta}}{dt} + j\omega_{me}\Lambda_{mg}e^{j\theta_{me}} \qquad (0.28)$$

And passing in a rotating reference system we obtain:

$$\boldsymbol{u}_{dq} = R\boldsymbol{i}_{dq} + L\frac{d\boldsymbol{i}_{dq}}{dt} + j\omega_{me}L\boldsymbol{i}_{dq} + j\omega_{me}\Lambda_{mg} \qquad (0.29)$$

The components are:

$$u_d = Ri_d + L\frac{di_d}{dt} - \omega_{me}Li_q \qquad (0.30)$$

$$u_q = Ri_q + L\frac{di_q}{dt} + \omega_{me}Li_d + \omega_{me}\Lambda_{mg} \qquad (0.31)$$

If we want to do a power balance we have to remember that the space vector is not conservative for the power, but it needs of a correction factor 3/2, so we have that the total power is:

$$p = \frac{3}{2} * \left(u_d i_d + u_q i_q\right) \qquad (0.32)$$

And substituting

$$p = \frac{3}{2} * \left( R\left(i_d^2 + i_q^2\right) + L\left(\frac{d\frac{i_d^2}{2}}{dt} + \frac{d\frac{i_q^2}{2}}{dt}\right) - \omega_{me} L i_q i_d + \omega_{me} L i_d i_q + \omega_{me} \Lambda_{mg} i_q \right) \qquad (0.33)$$

Where the Joule losses are:

$$p_{Joule} = \frac{3}{2} R\left(i_d^2 + i_q^2\right) \qquad (0.34)$$

The variation of magnetic energy is:

$$\frac{dW_m}{dt} = \frac{3}{2} L\left(\frac{d\frac{i_d^2}{2}}{dt} + \frac{d\frac{i_q^2}{2}}{dt}\right) \qquad (0.35)$$

And the mechanical power is:

$$p_m = \frac{3}{2}\left(-\omega_{me} L i_q i_d + \omega_{me} L i_d i_q + \omega_{me} \Lambda_{mg} i_q\right) \qquad (0.36)$$

$$p_m = \frac{3}{2}\left(\omega_{me} \Lambda_{mg} i_q\right) \qquad (0.37)$$

From the last expression we want find the torque equation. Remembering that the mechanical power is equal to the product between the torque and the mechanical speed of the motor we can write:

$$p_m = m * \omega_m \qquad (0.38)$$

But

$$\omega_{me} = p * \omega_m \qquad (0.39)$$

Where $p$ is the number of couples of poles, so we have:

$$p_m = m * \omega_m = \frac{3}{2}\left(\omega_m p \Lambda_{mg} i_q\right) \qquad (0.40)$$

Well the final expression of the torque is:

$$m = \frac{3}{2} p \Lambda_{mg} i_q \qquad (0.41)$$

For the IPM motors the magnetic field lines, made by the D current, cross the permanent magnets that have a magnetic permeability similar to the air one, therefore with a high reluctance, while the magnetic field lines made by the Q current don't cross the magnets, but they cross only the iron and the air gap, so the total reluctance of the magnetic circuit is lower than the other one.

$$\mathcal{R}_d > \mathcal{R}_q \qquad (0.42)$$

Well

$$L_d < L_q \qquad (0.43)$$

The equations in a rotating reference system are:

$$u_d = Ri_d + L_d \frac{di_d}{dt} - \omega_{me} L_q i_q \qquad (0.44)$$

$$u_q = Ri_q + L_q \frac{di_q}{dt} + \omega_{me} L_d i_d + \omega_{me} \Lambda_{mg} \qquad (0.45)$$

So doing the same previous calculations to find the expression of the torque, we obtain:

$$m = \frac{3}{2} p i_q \left(\Lambda_{mg} + \left(L_d - L_q\right) i_d\right) \qquad (0.46)$$

The scheme of the model is the following

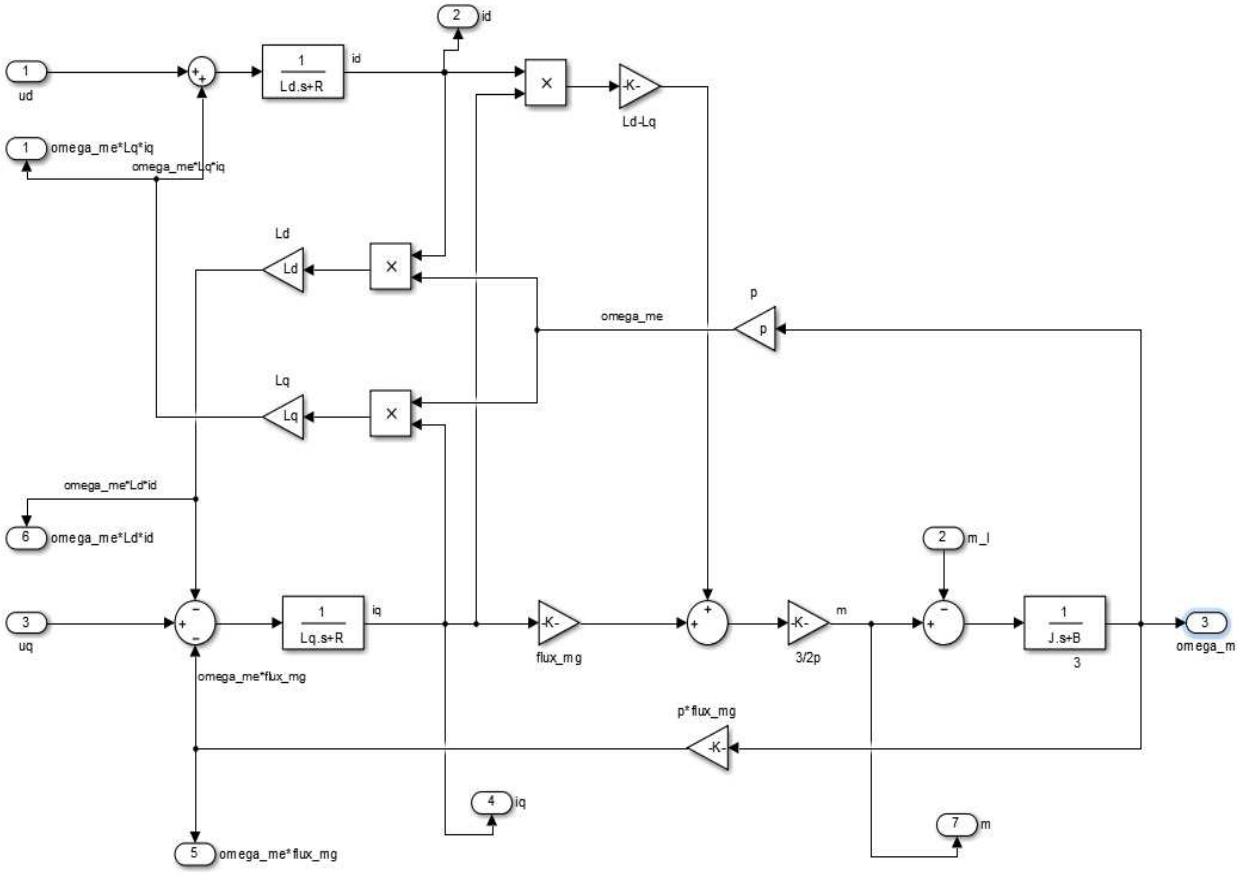


Figure 0.1

***Electric actuator***

An actuator is a component of machines that is responsible for moving or controlling a mechanism or system. An actuator requires a control signal and source of energy. The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power. The supplied main energy source may be electric current, hydraulic fluid pressure, or pneumatic pressure. When the control signal is received, the actuator responds by converting the energy into mechanical motion. An actuator is the mechanism by which a control system acts upon an environment. The control system can be simple (a fixed mechanical or electronic system), software-based (e.g. a printer driver, robot control system), a human, or any other input. An electric actuator is powered by a motor that converts electrical energy into mechanical torque. The electrical energy is used to actuate equipment such as multi-turn valves. It is one of the cleanest and most readily available forms of actuator because it does not involve oil.

The base structure of an electric actuator for a PM motor is presented in the following picture:

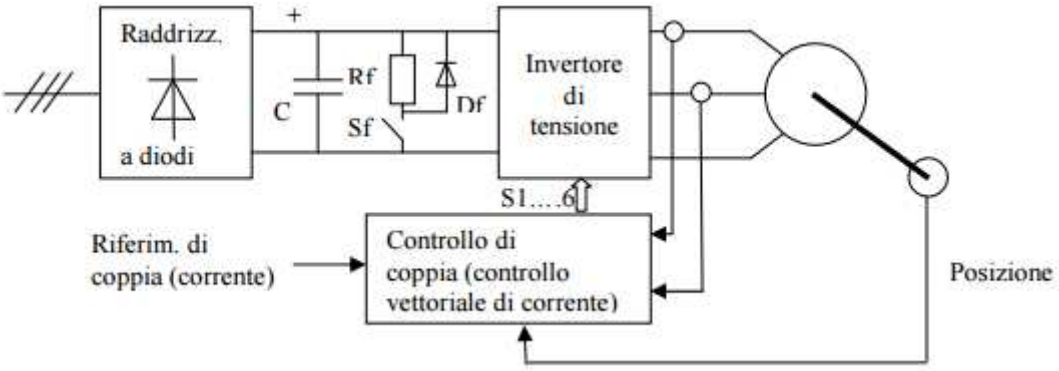


Figure 0.2

In the followings chapters will be presented the complete control proposed for the FACRI motor step by step.

# Chapter 1:
# The FACRI Project

### *1.1 Introduction*

In this chapter we will present the FACRI project in order to better understand what are the specifics required for the control and all the parameters of the DC-link, inverter and motor, that are necessary for our simulations.

### *1.2 The FACRI project*

Let's give some information about the FACRI project and the FACRI motor in particular. This project looks into the design of an actuator drive system required to meet the specifications given by FACRI in an effort to develop a reliable and high power density drive system which is optimised holistically rather than each component designed individually . This leads to an optimised solution at system level. This trade-off study deals with the concept selection of the electrical machine, the power electronics (PE) and the control scheme and identifies the key aspects to look at in the next design phase. These are dealt with individually within this report however consideration to the adjacent system components is given. After the initial concepts are selected some initial design work was done to confirm the feasibility of the solution proposed and identify the major aspects to consider and evaluate in the subsequent preliminary design phase. This considers the electromagnetic, mechanical, power conversion and thermal aspects required. The key objectives of the trade-off study are:

a) Identify potential machine topologies and quantitatively select the most promising to take to the preliminary design stage

b) Identify potential converter configurations and power device technologies and quantitatively select the most promising to take to the preliminary design stage.

c) Identify the main thermal management solutions and quantitatively down select the most promising.

d) Identify the most promising control platform to enable the development of the drive system. [ this point in particular consists in our work ]

e) Identify the most impacting parameters in the system and within each constituent component to ensure an optimisation design process which captures these parameters and ensures an overall optimised solution.

f) Determine a structure for the preliminary design of the drive system

***Output requirements***

For sake of completeness, the summary of the project requirements is reported in this section.

1 Without oil-cooling, the system needs to be able to achieve a linear torque-speed curve, where the stall torque equals to 3.4Nm and a continuous torque equal to 1.6Nm when the speed is 19000 rpm (3.18kw). (Room temperature, installed in test frame and no forced cooling). This condition is not an application requirement but rather a guideline for isolated motor-drive testing within a laboratory environment.

2 With oil cooling, a 10.5Nm continuous torque is to be delivered within a speed range of 0 to 8700 rpm (9.6kw) and a continuous torque of 5Nm at 19000 rpm (9.9kw). This is expected to be the normal working condition of the drive in the application environment.

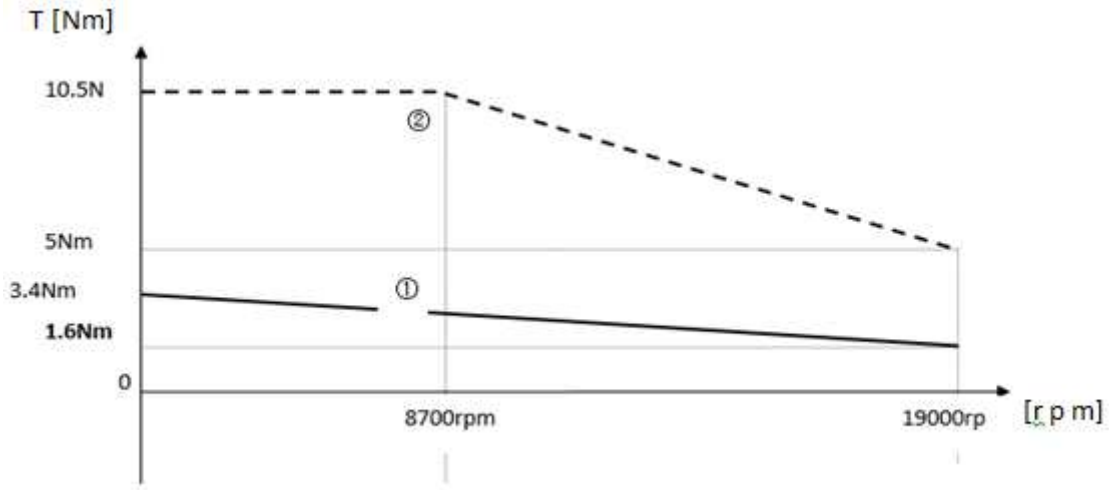


Figure 1.1 : Torque-speed characteristic

***Dimensional constraints***

A. Motor axial cross section :≦80mm×80mm

B. Weight ≦ 2.5Kg, including all active parts: rotor, stator, front/ending bearing, resolver and testing frame, excluding the cooling fluid mass.

***Thermal management***

The electrical machine is to be fluid cooled (oil) with properties to be determined within the preliminary design phase and the power electronics will be naturally ventilated.

### *1.3 The motor*

Now let's focus on the motor of the project. Permanent magnet (PM) motors are electro-mechanical devices where the field is achieved by the use of PMs, instead of the more traditional wound field coils. This results in simplification in construction, reduction in losses and improvement in efficiency. PMSMs, which are widely used in variable speed drive applications, are synchronous machines with excellent performances in terms of power and torque density and efficiency. PMSMs can be loosely classified into two main groups, relative to the placement of the PMs on the rotor, namely SMPM machines and IPM motors.

For SMPM motors, the PMs are mounted on the outer surface of the rotor as shown in Figure. This arrangement provides higher torque density due to high air gap flux density, leading to a potential minimization of machine volume and mass. An SMPM motor also enjoys lower rotor losses and thus higher efficiency. The main demerits of this configuration are the risk of PM demagnetization, and the need for retaining sleeves for the PMs especially for high speed machines as in the application considered here. The rotor has an iron core that may be solid or may be made of laminations. An important aspect of SMPM machines is that it is a non-salient machine, resulting in that direct axis inductance $L_d$ and the quadrature axis inductance $L_q$ of the PMSM are equal (Ld=Lq).

IPM motors have their PMs "buried" inside the rotor as shown in Figure. This configuration typically weakens the mechanical strength of the rotor however somewhat provides for magnet demagnetization protection. The motors cannot be used for very high speed applications since the magnets are only physically contained using magnetic steel which is often not very mechanically robust. In addition they typically have lower critical speeds compared with the surface mount counterpart. Torque density performance of such a machine is comparable (if slightly lower) to that of an SMPM motor for the same kVA rating, while torque ripple is somewhat higher. These motors have a saliency effect with q axis inductance greater than the d axes inductance (Lq > Ld) if designed properly. It is important to highlight however that the extent of this saliency is however very dependent on the winding configuration, pole number and air gap clearance.

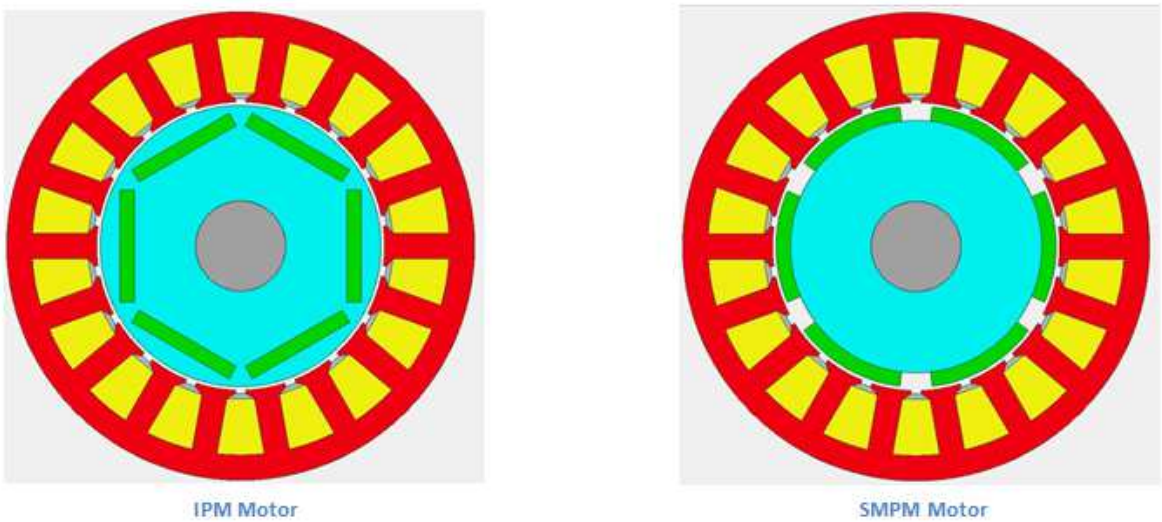


Figure 1.2

An important advantage of PM machines is that they are synchronous machines in which the excitation is provided by the magnets on the rotor. In distributed-wound versions of these machines, the magnet losses are low, and can be lowered to negligible values by magnet segmentation. This can result in thermal management benefits as cooling rotating components can be very challenging due to their naturally high resistance thermal path to ambient. Unlike IMs, PM machines are well suited for design at higher pole numbers if the power electronics switching allows. This can result in advantages in terms of having lower inertia rotors as well as rotors with significantly reduced weight (hollow rotors).

PM machines offer great flexibility in terms of design when compared to induction machines as they are easier to adopt higher pole numbers as discussed above, different winding structures, lower inertia rotors and more importantly larger air gaps when compared to all other machines. In recent detailed study the authors of this report did within the clean sky framework it was found out that if a high peak to rated (continuous) torque is required then PM machines perform better for actuation applications. A perceived drawback of PM machines is their behavior under faulty conditions especially in safety-critical environments. The issues here range from the possibility of uncontrolled fault currents in the event of a short-circuit fault to the potential large terminal voltages due to the magnet flux in case of a converter malfunction at high speed. Whilst these drawbacks are important, there are a number of ways they can be managed by design and control.

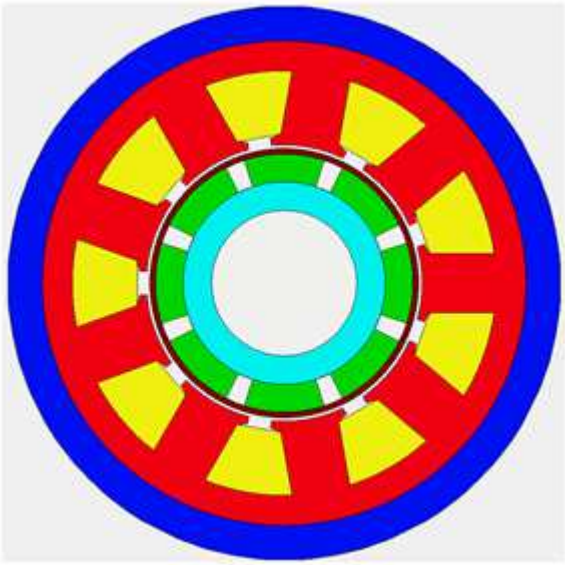

Figure 1.3 : Cross-section of 8p/9s SPMSM

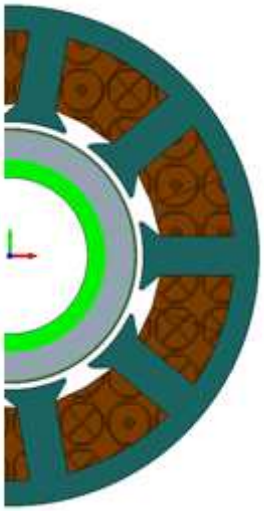

Figure 1.4 : Quasi-Halbach array

The last figure represents the cross section of the final model of our motor.

In the following tables there are the machine design dimensions:

| PM machine with CoFe.1 (VacoFlux 48) and Quasi-Halback array | | |
|---|---|---|
| | | |
| | N Coil | 12 |
| | Dc link voltage | 270 V |
| | Peak current | 78 A |
| | | |

Table 1.1

| | |
|---|---|
| Stack Length | 80 mm |
| Air-gap thickness (g) | 0.9 mm |
| Sleeve thickness ($g_{sleeve}$) | 0.4 mm |
| Split ratio | 0.535 |
| Stator inner diameter | 37.45 mm |
| Rotor outer diameter | 35.65 mm |
| Rotor Back Iron | 2.5 mm |
| Tooth width | 5.5 mm |
| Slot Depth | 12.775 mm |
| Stator back iron thickness | 3.5 mm |
| Slot opening | 3.75 mm |
| Tooth Tip | 0.5 mm |
| Tooth Wedge Height | 2 mm |
| Magnet Thickness | 4 mm |
| Fill factor (FF) | 0.36 |

Table 1.2

While here is reported the main code in Matlab that contains all parameters of our control model.

```
%%%%  FACRI motor parameters

speed_rpm = 19000;                  %operating speed [rpm]
omega_m=speed_rpm/60*2*pi;          %operating speed [rad/s]
p=4;                                %poles
R=0.0951;                           %phase resistance [ohm]
Ld = 211e-6;                        %d axes inductance [H]
Lq = 306e-6;                        %q axes inductance [H]
B=0.0004432;                        %viscosity friction [Nms]
J=1e-4;                             %momento di inerzia [kgm^2]
flux_mg=0.0236;                     %[Vs]
U_lim_LL=270;                       %Line to Line voltage limit [V]
U_lim=U_lim_LL/sqrt(3);             %phase voltage limit [V]
I_lim=78;                           %phase current limit [V]
m_max = 12;                         %maximum torque [Nm]
```

```
%%%% converter parameters

Fs = 16e3;                          %switching frequency [Hz]
Ts = 1/Fs;                          %switching period [s]
Fs_boost = 1;
Vdc = 270;                          %DC-link voltage [V]
Vdc_up_precharge = Vdc/2;           % [V]
Vdc_dw_precharge = Vdc/2;           % [V]
Cdc_up = 1200e-6;                   % [F]
Rcdc_up = 10e-6;                    %[ ohm]
Cdc_dw = 1200e-6;                   % [F]
Rcdc_dw = 10e-6;                    % [ohm]
t_dead= 2.5e-6;                     %dead time [s]
```

we can observe that the d and q axes inductances are different, so in the control we have implemented an IPM motor.

Initially the motor was implemented in Simulink with the scheme showed in the introduction, but when the control was increasing its completeness we needed of a motor implemented in Plecs, that is another Matlab tool, because was the only way to use also the inverter implemented with Plecs which permits to calculate a lot of losses (switching losses, thermal losses, ecc). In the Plecs library there is already a model of a PM machine but we had to give up to use it because the references that it uses for the Park transform in the C-script are different to ours. So we built a circuital model in d-q inside a Plecs circuit to implement our motor like showed in the following figure.

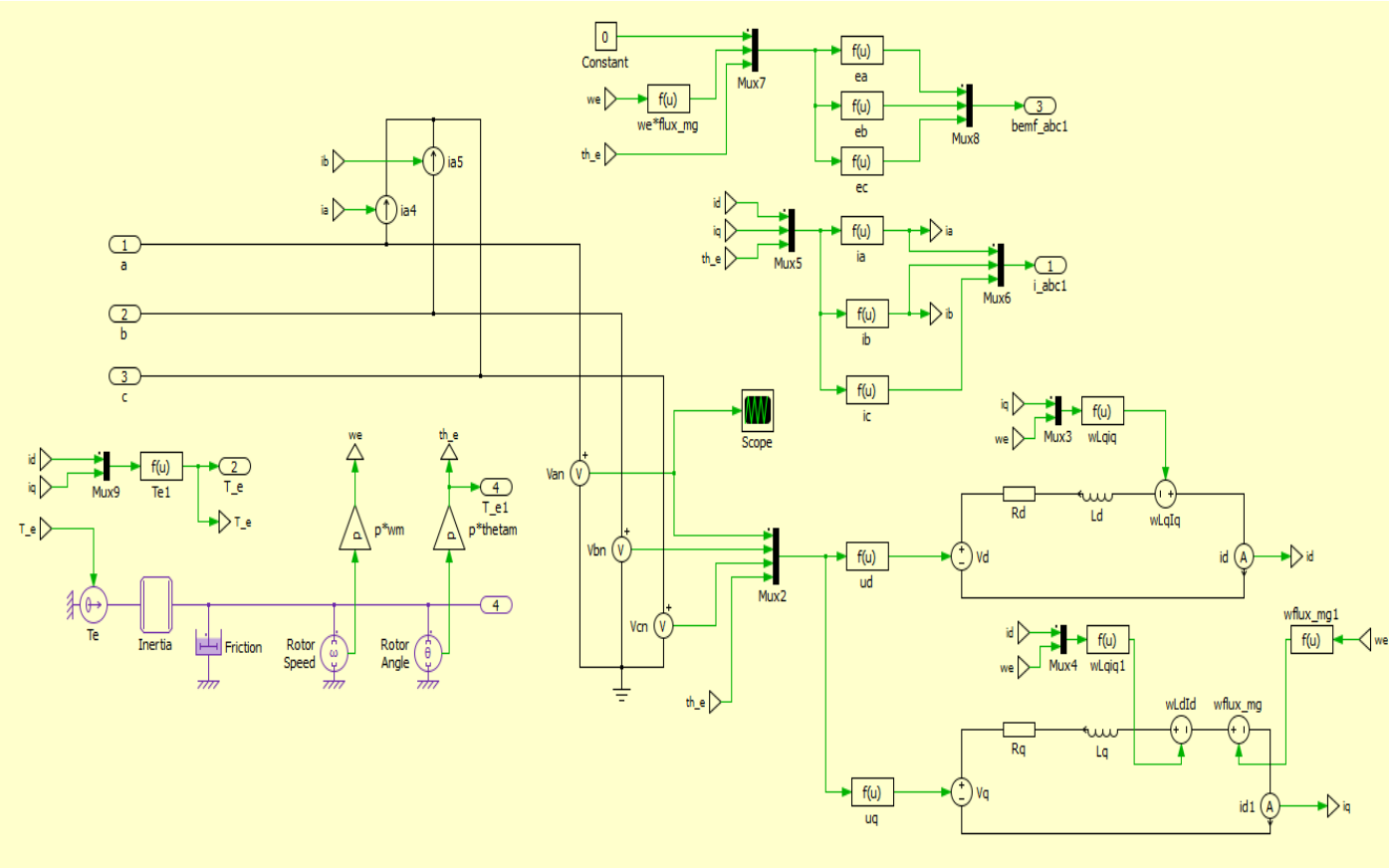


Figure 1.5 : Circuital model of the motor

The yellow background means that it is a Plecs circuit block. Plecs is useful because it runs C-scripts, while Simulink solves M-scripts, in this way we make the simulation faster, in fact the C-code is very faster than the M-code. Black lines connect electrical components, green lines transport signals and the violets are for the mechanical components of our model like the inertia, the friction and the torque load. The inputs of the motor are the outputs of the inverter and the outputs of the motor are the phase currents, the torque, the back electromotive forces and the electric position of the rotor like we can observe in the figure below, where are presented also the modulator and the inverter.

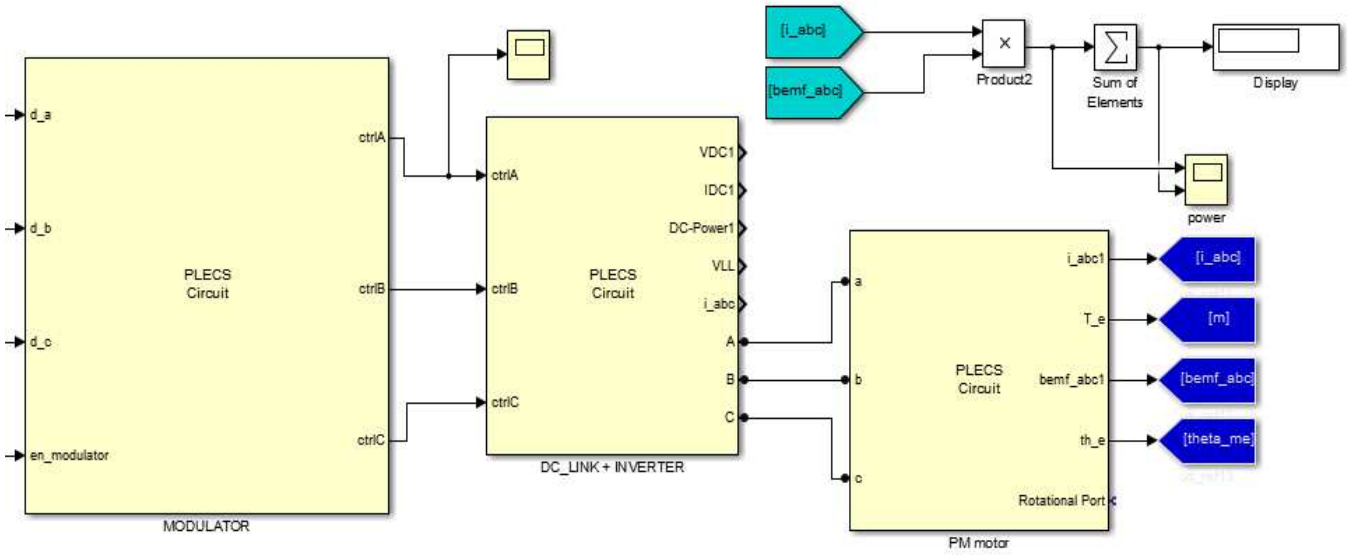


Figure 1.6 : Plecs blocks

### *1.4 The modulator and the inverter*

In the first simulation we have considered the modulator and the inverter only like a delay equal to 1.5*Ts, where Ts is the sampling period of the PWM modulator equal to 1/Fs, and Fs is the switching frequency. The modulator is approximated like a delay produced by the calculation time required to the elaboration of the signals for the inverter. This delay is about Ts, so the inverter will receive the control signals with a delay equal to a sampling period:

$$G(s)_{modulator} = e^{-sT_s} = \frac{1}{e^{sT_s}} \qquad (1.1)$$

and if we use Taylor until the first order we have:

$$G(s)_{modulator} = \frac{1}{e^{sT_s}} \cong \frac{1}{1 + sT_s} \qquad (1.2)$$

Now let's see how is done the transfer function of the inverter:

$$G(s)_{inverter} = \frac{1}{1 + s\frac{T_s}{2}} \qquad (1.3)$$

well the total delay produced by the inverter and the PWM is Ts+Ts/2, so we obtain:

$$G(s)_{PWM+inverter} = e^{-s\frac{3}{2}T_s} \cong \frac{1}{1 + s\frac{3}{2}T_s} \qquad (1.4)$$

always using Taylor until the first order.

Using the technique of the axes decoupling, we can write the last transfer function for the d and q axes both like we can see in the following figure.

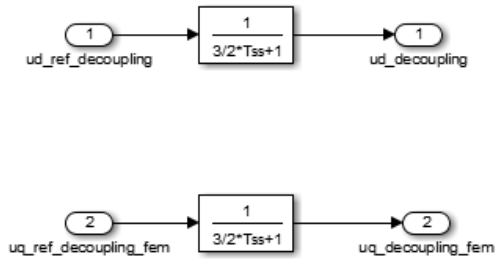


Figure 1.7

Once done the ideal simulation, the next step of the work was considering a real modulator and a real inverter with the switching, the dead times, the DC-link, and all the losses that the simple delay that we had before didn't consider. In the figure 1.6 we observe that the inputs of the modulator are the abc voltage references normalized with half DC-link, while in the previous the inputs were the voltages in the dq reference, so we need of a transformation to move from a dq reference to a abc reference. In the figure we can observe the transformation and the normalization of the voltages.

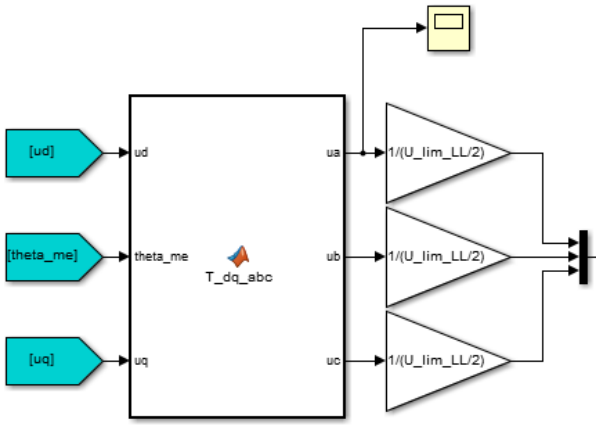


Figure 1.8

The Matlab code for the transformation is showed in the appendix of this chapter.

The modulator makes a comparison between the input signals and a carrier included between -1 and 1, then it takes the result and produces a vector which the second component is the negative of the first. The modulator produces three vectors, one for each inverter leg, in fact we have a three phase inverter. In the following figure we can see the block of the modulator, made with PLECS, and under its mask.

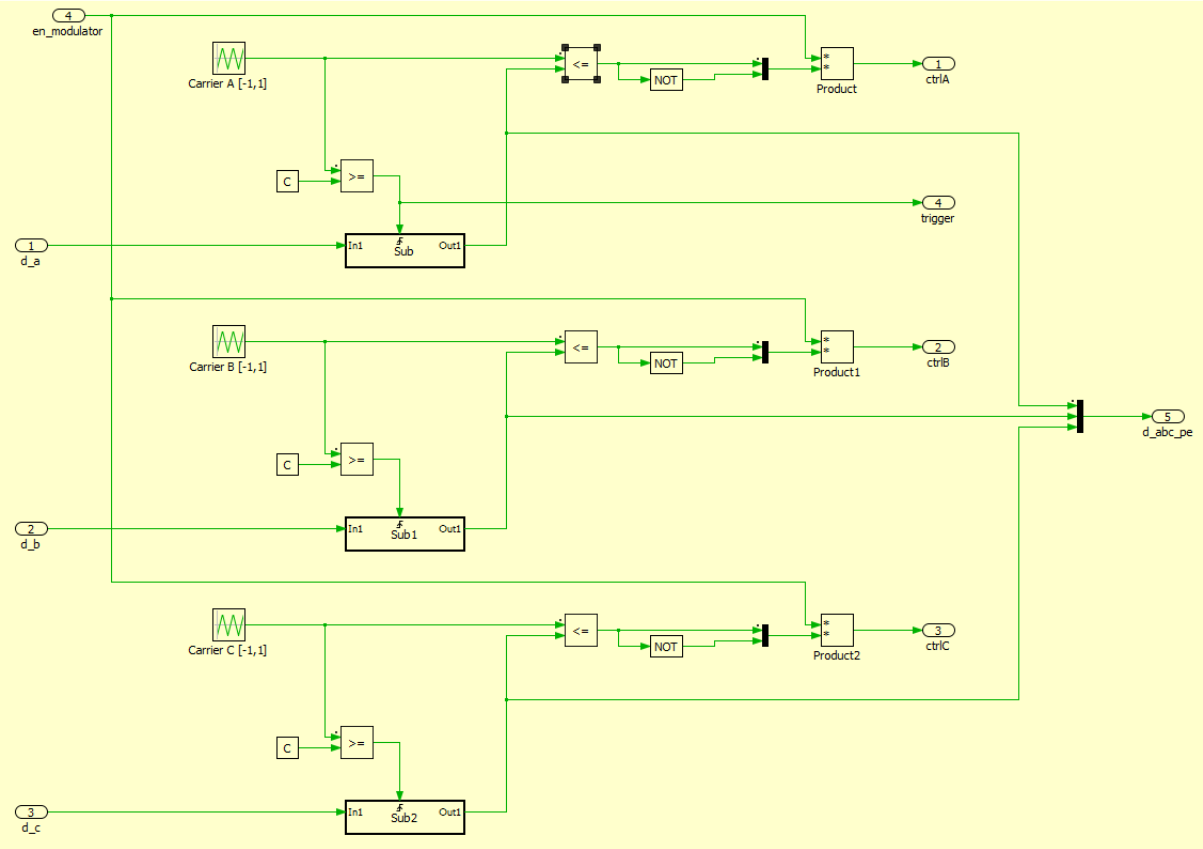


Figure 1.9

The inverter is obviously represented with the DC-link that is the alimentation of the motor. In the figure we can see the circuit of the DC-link, where there are the line to line voltage source V_dc of 270 [V] , two resistors and two capacitors of 10 [μΩ] and 1200 [μF] respectively. We have used also a voltmeter and an ammeter to measure the voltage, the current and the power of the DC-link.

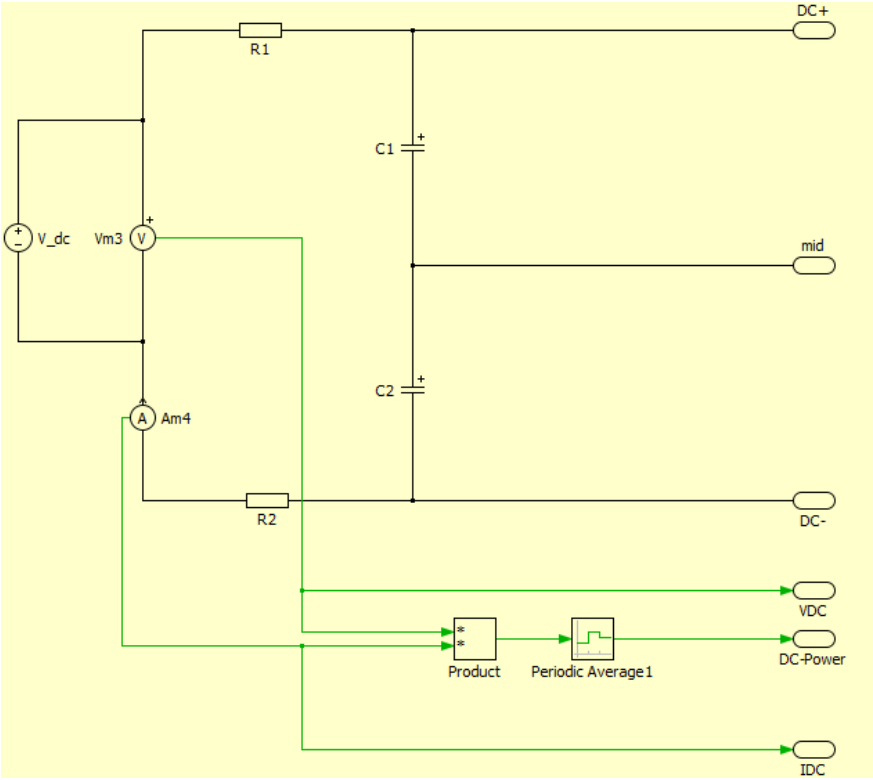


Figure 1.10

Now let's observe how the inverter is made. It is connected with the modulator and with the DC-link, it is a three phase inverter, like said before, and every leg is composed by two IGBT transistors and two diodes. In the following figure we can see under all the masks of the block of the inverter made with PLECS.

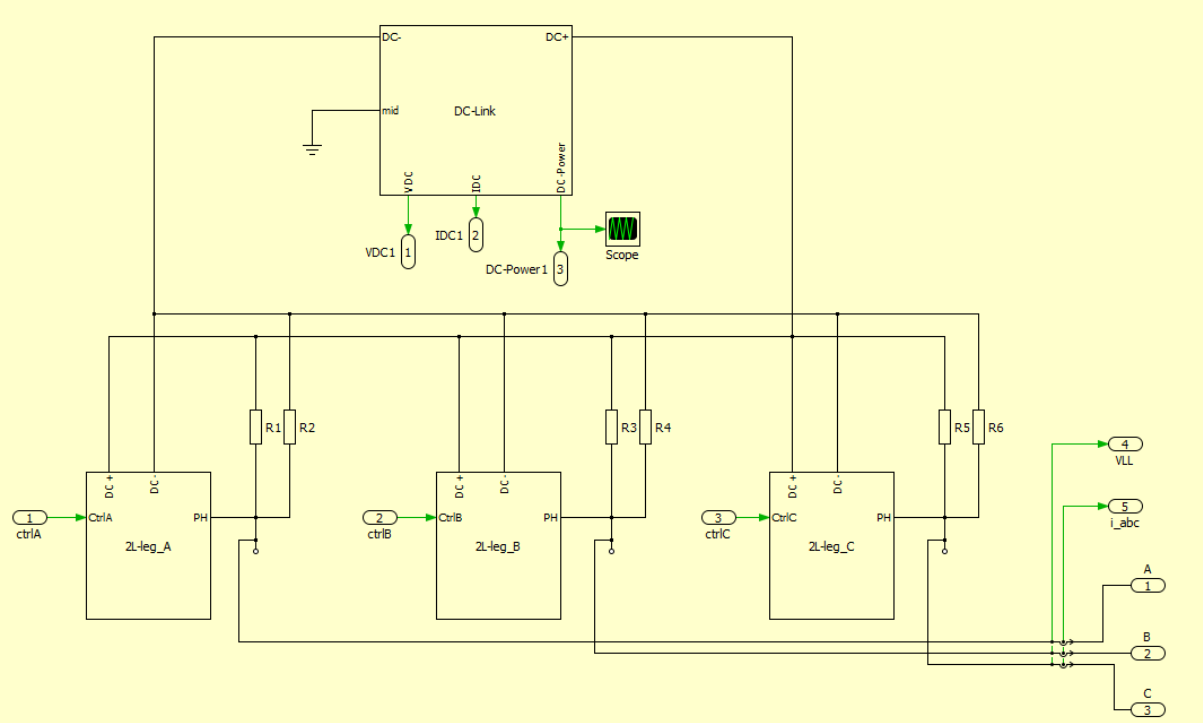


Figure 1.11 : DC-link and three phase inverter

A curiosity: the resistors R1 R2 R3 R4 R5 R6 are of the order of mega ohm well it's like they aren't, but they are of help to smooth all the discontinuities made by the switches, that are damaging for the software calculations. Now let's see inside one of the legs.

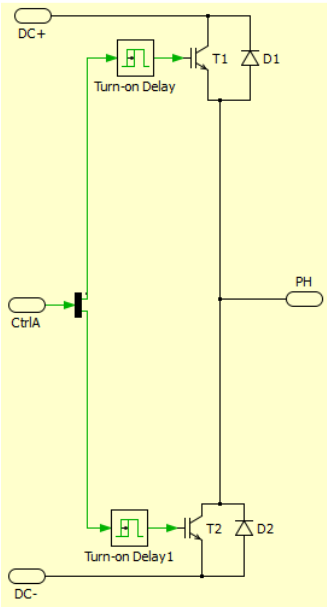


Figure 1.12 : leg A

In the last figure we can observe that the input ‘’ctrlA’’ which comes from the modulator, it is corrected by two blocks that make a delay of 2.5 [µs]. This particular delay represents the dead time for the switches, so we are sure to avoid a short circuit with the DC-link.

Considering the modulator and the inverter we have to discretize all the Simulink model and the simulations are very different respect considering them like a simple delay, and it is more difficult control the full system.

### *1.5 Appendix*

Transformation from d-q to abc:

```
function [ua,ub,uc] = T_dq_abc(ud,theta_me,uq)


%%%% T_dq_abc


 T1=[cos(theta_me) sin(theta_me)
    -sin(theta_me) cos(theta_me)];  % T_allfabeta_dq

 T2=inv(T1);                          % T_dq_alfabeta
 T3=[1 0 1
    -0.5 sqrt(3)/2 1
    -0.5 -sqrt(3)/2 1];             % T_alfabetazero_abc

 g_0=0;                               % componente omopolare
 u_alfabeta=T2*[ud
    uq];
 u_alfabeta0=[u_alfabeta
    g_0];
 u_abc=T3*u_alfabeta0;
 ua=u_abc(1);
 ub=u_abc(2);
 uc=u_abc(3);
```

# Chapter 2:

# Control algorithm

### *2.1 Introduction*

In this chapter will be presented the structure of our control in order to satisfy the requirements of the project, in particular will be explained in detail the MTPA (Max Torque Per Ampere), the FW (Flux Weakening) algorithms and PI regulators used in the two current loops.

### *2.2 Control scheme*

The test of the motor in the lab will be done with another motor that holds the speed, therefore our control scheme is been projected without the speed loop and only the two current loops. In the following figure is showed an example of current control.

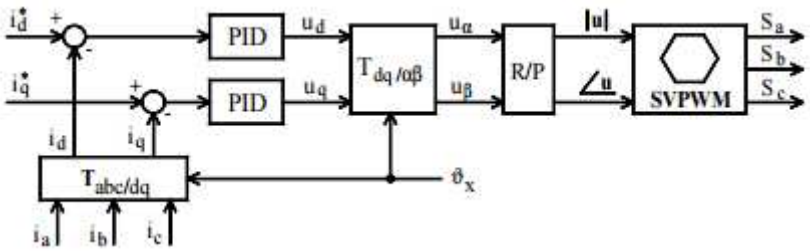


Figure 2.1

$i_d^*$ and $i_q^*$ are the references given by the MTPA and FW algorithm, known the torque-speed characteristic requirement, and the PID are the (Proportional Integral Derivative) regulators, but in our case normal PI will be enough. Now let's see how the two references $i_d^*$ and $i_q^*$ are generated.

### *2.3 MTPA*

Now we are considering the steady state, so all the time derivative terms in the equations seen in the introduction will be neglected and will be used a capital letters notation for the physical quantities.

The operation limits for an IPM machine are the followings:

$$\begin{cases} I_d^2 + I_q^2 \leq I_{lim}^2 \\ U_d^2 + U_q^2 \leq U_{lim}^2 \end{cases} \qquad (2.1)$$

But in steady state we have:

$$\begin{cases} U_d = RI_d - \Omega_{me} L_q I_q \\ U_q = RI_q + \Omega_{me}(L_d I_d + \Lambda_{mg}) \end{cases} \qquad (2.2)$$

So, if we substitute the last in the previous system and we neglect the resistivity terms, we obtain:

$$\begin{cases} I_d^2 + I_q^2 \leq I_{lim}^2 \\ U_d^2 + U_q^2 = \Omega_{me}^2 L_q^2 I_q^2 + \Omega_{me}^2 (L_d I_d + \Lambda_{mg})^2 \leq U_{lim}^2 \end{cases} \qquad (2.3)$$

The first equation is the current limit and it is the equation of a circumference, while the second one corresponds to the voltage limit and it is the equation of an ellipse, like we can observe in the bottom figure.

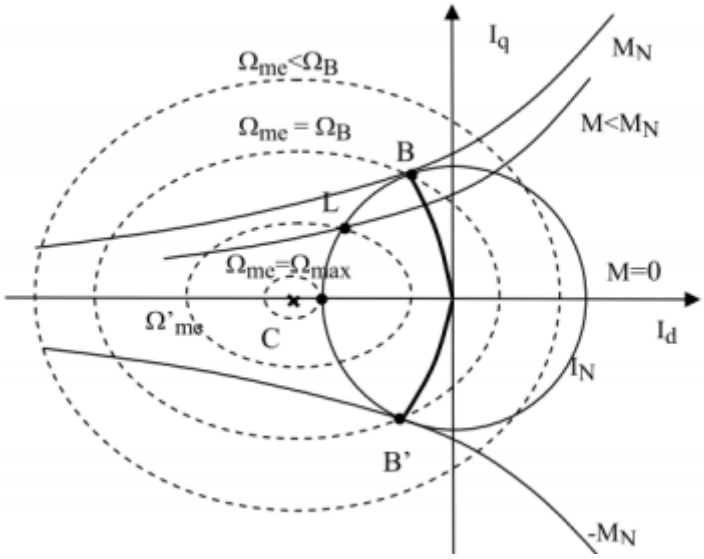


Figure 2.2

Our purpose is to find the currents *Id* and *Iq* of the MPTA (Max Torque Per Ampere) condition, knowing the torque reference *m_ref*. To do this, we need of the relation between the torque and the currents in MTPA. The torque equation is the following:

$$M = \frac{3}{2} p\left[\Lambda_{mg} + \left(L_d - L_q\right) I_d\right] I_q \qquad (2.4)$$

But if we write the *Iq* in function of M we have:

$$I_q = \frac{M}{\frac{3}{2} p\left[\Lambda_{mg} + \left(L_d - L_q\right) I_d\right]} \qquad (2.5)$$

that is the equation of the hyperbole in the figure 2.2. We have the MTPA condition if we impose the orthogonality between the tangent of the hyperbole and the straight line for the O, so writing with mathematical language we have the followings relations.

$$m_1 = \frac{dI_q}{dI_d} = \frac{2M}{3p} * \frac{-1}{\left[\Lambda_{mg} + \left(L_d - L_q\right) I_d\right]^2} * \left(L_d - L_q\right) \qquad (2.6)$$

and substituting the expression of the torque *M*:

$$m_1 = -\frac{I_q(L_d - L_q)}{\Lambda_{mg} + (L_d - L_q)I_d} \qquad (2.7)$$

But the angular coefficient of the straight line for the O is:

$$m_2 = \frac{I_q}{I_d} \qquad (2.8)$$

and the orthogonality relation is:

$$m_1 = -\frac{1}{m_2} \qquad (2.9)$$

So we obtain:

$$I_q = \pm\sqrt{\frac{I_d[\Lambda_{mg} + (L_d - L_q)I_d]}{L_d - L_q}} \qquad (2.10)$$

Now that we have the relation between the *Id* and *Iq* in MTPA, we can use it with the torque equation like follows:

$$\begin{cases} M = \frac{3}{2}p[\Lambda_{mg} + (L_d - L_q)I_d]I_q \\ I_q = \pm\sqrt{\frac{I_d[\Lambda_{mg} + (L_d - L_q)I_d]}{L_d - L_q}} \end{cases} \qquad (2.11)$$

Now let's make some calculations:

$$\begin{cases} I_d = \left[\frac{2M}{3pI_q} - \Lambda_{mg}\right]\frac{1}{L_d - L_q} \\ I_q^2 = \frac{\Lambda_{mg}I_d}{L_d - L_q} + I_d^2 \end{cases} \qquad (2.12)$$

Substituting *Id* in the second equation and skipping some calculations we have:

$$I_q^4 = \frac{4M^2 - 6M\Lambda_{mg}pI_q}{9p^2(L_d - L_q)^2} \qquad (2.13)$$

Therefore

$$-4 * M^2 + 6p\Lambda_{mg}I_q * M + 9p^2(L_d - L_q)^2 I_q^4 = 0 \qquad (2.14)$$

The last one is a second order equation that can be resolved analytically and the roots are the followings:

$$M_{1,2} = \frac{6pI_q\left[\Lambda_{mg} \pm \sqrt{\Lambda_{mg}^2 + 4(L_d - L_q)^2 I_q^2}\right]}{8} \qquad (2.15)$$

The solution with the ‘’minus’’ is not acceptable because even if *Ld=Lq* we have to have torque in any case, so we have obtained finally:

$$M = \frac{3}{4} p I_q \left[ \Lambda_{mg} + \sqrt{\Lambda_{mg}^2 + 4(L_d - L_q)^2 I_q^2} \right] \qquad (2.16)$$

that is the relation between the torque and the current *Iq* in the MTPA condition, but in our control the torque is the input and the *Iq* is the output, so we have to find *Iq=f(M)*.

A solution could be to use the function polyfit in Matlab to find a polynomial that approaches the function *Iq=f(M)*, and so we have done obtaining a good result, infact from the following figure 2.3 we can’t see the error, but we have to zoom a lot before to see the error of the approximation and, like we can see in the figure 2.4, the error can be neglect.

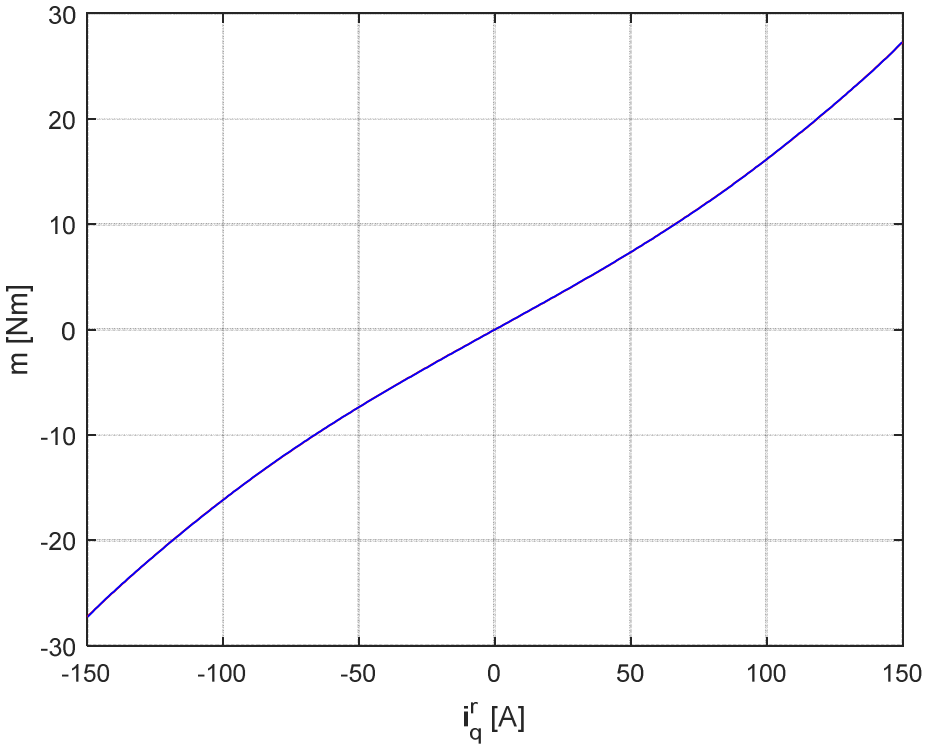


Figure 2.3 : equation 2.16 vs polyfit

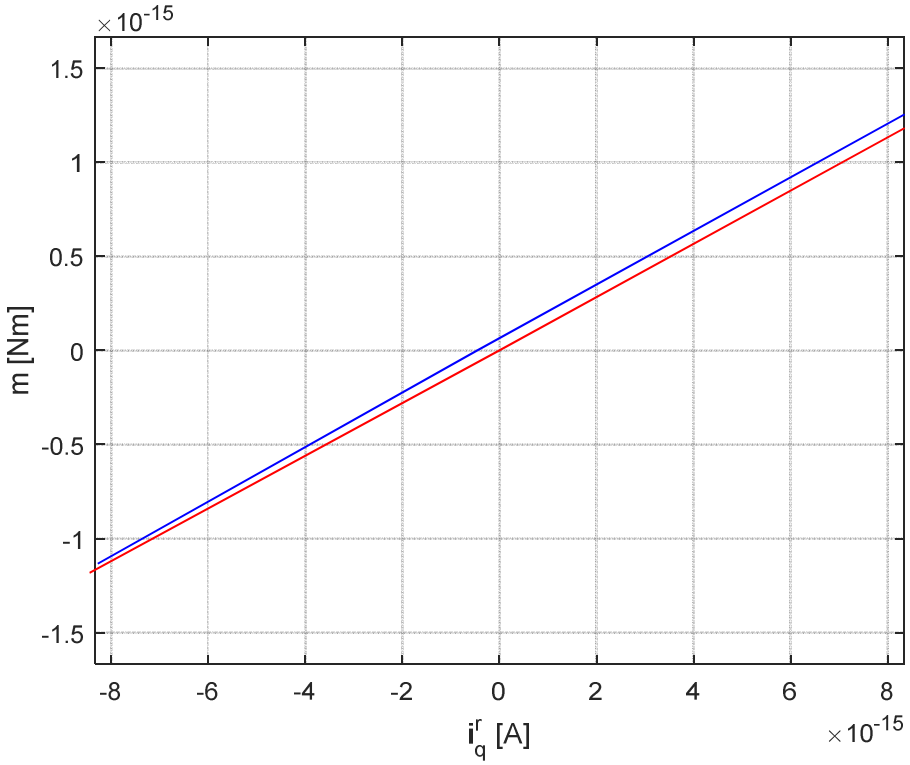


Figure 2.4

The function polyfit finds the coefficients *Cq*, well we can write the *Iq* like follows:

$$I_{q\_MTPA} = C_{q1}M^8 + C_{q2}M^7 + C_{q3}M^6 + C_{q4}M^5 + C_{q5}M^4 + C_{q6}M^3 + C_{q7}M^2 + C_{q8}M^1 + C_{q9}M^0 \qquad (2.17)$$

and so we have obtained the relation between the current *Iq* and the torque *M* in the MTPA condition that gives us the *Iq_MTPA* in output from the input of the torque *M.* Now that we know *Iq_MTPA* it's easy to find the *Id_MTPA* because we only have to substitute it in the *Iq* equation of 2.11 and we obtain:

$$I_{q\_MTPA}^2 = \frac{\Lambda_{mg} I_{d\_MTPA}}{L_d - L_q} + I_{d\_MTPA}^2 \qquad (2.18)$$

That is a second order equation where the unique unknown is *Id_MTPA*, well we have:

$$\Delta = \frac{\Lambda_{mg}^2}{(L_d - L_q)^2} + 4I_{q_{MTPA}}^2 \qquad (2.19)$$

$$I_{d\_MTPA1,2} = \frac{-\frac{\Lambda_{mg}}{L_d - L_q} \pm \sqrt{\Delta}}{2} \qquad (2.20)$$

but we want a negative *Id_MTPA* so our solution to implement is:

$$I_{d\_MTPA} = \frac{-\frac{\Lambda_{mg}}{L_d - L_q} - \sqrt{\Delta}}{2} \qquad (2.21)$$

Finally we can use the last two equations to implement in the MTPA block in Simulink for our control. In the appendix of this chapter there is the code that we have used in the main to calculate the coefficients vector which we need inside the MTPA block in Simulink.

### *2.4 FW*

When the motor speed increases, the voltage limit becomes smaller like we can observe from its equation:

$$\left(\frac{L_q}{L_d}\right)^2 I_q^2 + \left(I_d + \frac{\Lambda_{mg}}{L_d}\right)^2 \leq \frac{U_{lim}^2}{L_d^2 \Omega_{me}^2} \qquad (2.22)$$

Where

$$\Omega_{me} = p\Omega_m \qquad (2.23)$$

and *Ωm* is the rotor speed, so it's possible that if we work always in the MTPA condition, we don't respect the voltage limit with the increasing of the speed, therefore we have to work in Flux Weakening. Working in FW means that our operation point is moving on the circumference of the current limit like showed in the bottom figure.

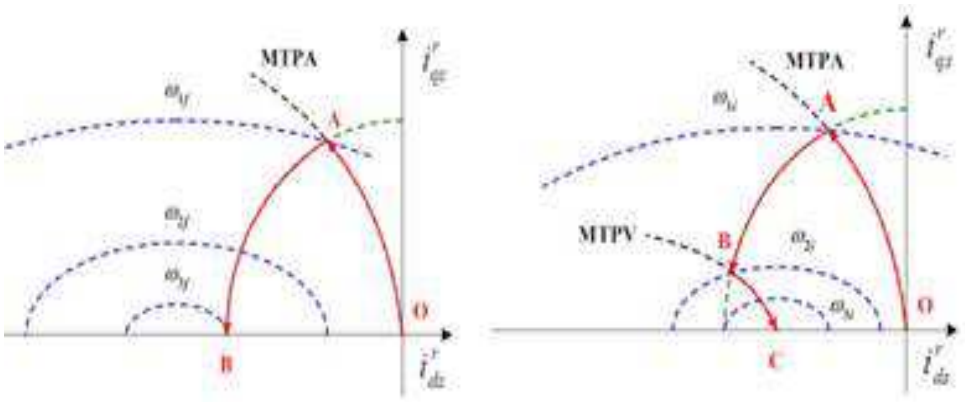


Figure 2.5

The second image shows that if the centre of the voltage limit is inside the circumference of the current limit, we have to work in FW until a certain speed and after that we have to work in the MTPV (Max Torque Per Voltage) condition, but it's not our case. Below there are our case curves of the current limit (in red) and of the voltage limit neglecting the resistance and vary the speed (in blue).

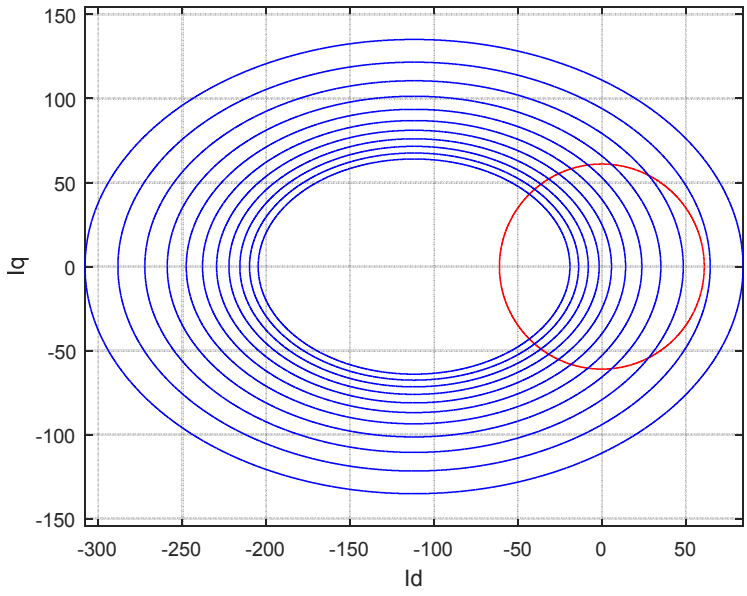


Figure 2.6 : voltage and current limits

To implement the Flux Weakening, we have used a simple ''if loop'' in Matlab where we give the followings instructions:

If $$\left(\frac{L_q}{L_d}\right)^2 I_{q\_MTPA}^2 + \left(I_{d\_MTPA} + \frac{\Lambda_{mg}}{L_d}\right)^2 \leq \frac{U_{lim}^2}{L_d^2 \Omega_{me}^2} \qquad (2.24)$$

$$I_{d\_FW} = I_{d_{MTPA}} \qquad (2.25)$$

$$I_{q\_FW} = I_{q\_MTPA} \qquad (2.26)$$

Else

$$\begin{cases} \left(\frac{L_q}{L_d}\right)^2 I_{q\_FW}^2 + \left(I_{d\_FW} + \frac{\Lambda_{mg}}{L_d}\right)^2 = \frac{U_{lim}^2}{L_d^2 \Omega_{me}^2} \\ I_{q\_FW}^2 + I_{d\_FW}^2 = I_{q\_MTPA}^2 + I_{d\_MTPA}^2 \end{cases} \qquad (2.27)$$

End

To resolve the system, we find *Id_FW* first and later *Iq_FW*. Substituting the second equation in the first in 2.27 and making some calculations we obtain:

$$\left[1 - \left(\frac{L_q}{L_d}\right)^2\right] I_d^2 + \left(\frac{2\Lambda_{mg}}{L_d}\right) I_d + \frac{\Lambda_{mg}^2}{L_d^2} - \frac{U_{lim}^2}{L_d^2 \Omega_{me}^2} + \left(\frac{L_q}{L_d}\right)^2 * \left(I_{q_{MTPA}}^2 + I_{d_{MTPA}}^2\right) = 0 \qquad (2.28)$$

That is a simple second order equation with a ''delta'' equals to:

$$\Delta = \left(\frac{2\Lambda_{mg}}{L_d}\right)^2 - 4\left[1 - \left(\frac{L_q}{L_d}\right)^2\right] * \left[\frac{\Lambda_{mg}^2}{L_d^2} - \frac{U_{lim}^2}{L_d^2 \Omega_{me}^2} + \left(\frac{L_q}{L_d}\right)^2 * \left(I_{q_{MTPA}}^2 + I_{d_{MTPA}}^2\right)\right] \qquad (2.29)$$

Therefore

$$I_{d\_FW1,2} = \frac{-\left(\frac{2\Lambda_{mg}}{L_d}\right) \pm \sqrt{\Delta}}{2\left[1 - \left(\frac{L_q}{L_d}\right)^2\right]} \qquad (2.30)$$

Since we want an *Id<0*, so the numerator has to be positive because the denominator is always negative as *Lq>Ld*, therefore the right solution is:

$$I_{d\_FW} = \frac{-\left(\frac{2\Lambda_{mg}}{L_d}\right) + \sqrt{\Delta}}{2\left[1 - \left(\frac{L_q}{L_d}\right)^2\right]} \qquad (2.31)$$

Well

$$I_{q\_FW} = sign(M) * \sqrt{\left(I_{q_{MTPA}}^2 + I_{d_{MTPA}}^2\right) - I_{d_{FW}}^2} \qquad (2.32)$$

Finally we have built a block in Simulink that with the inputs of torque and speed, decides if working in MTPA or FW condition, and the outputs will be reference currents for the current loops of our control.

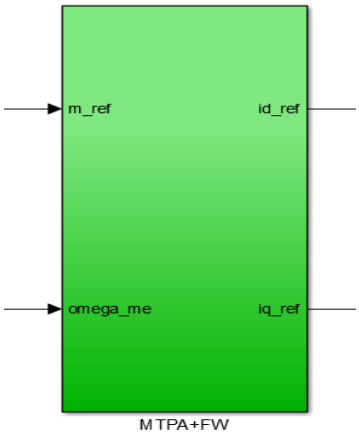


Figure 2.7

Under the mask there is the following scheme:

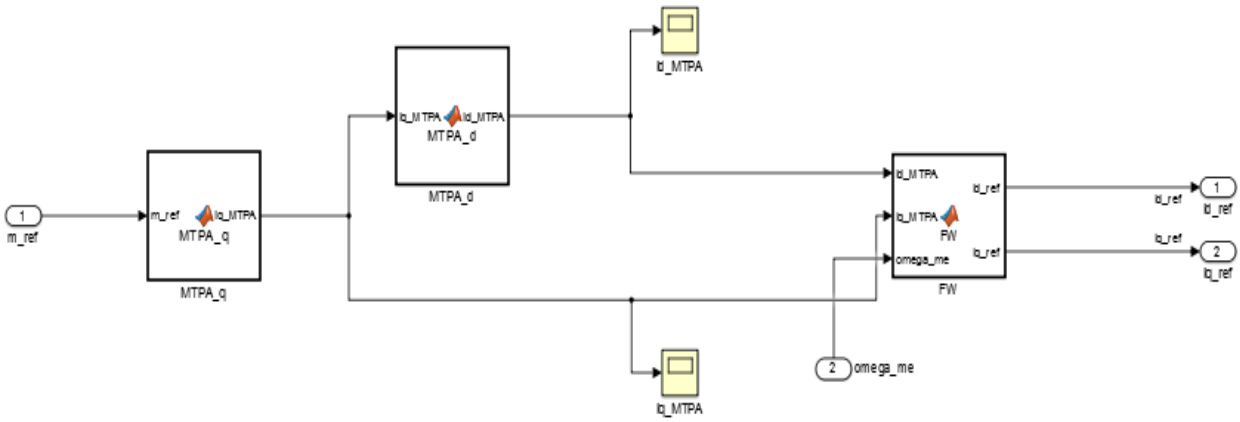


Figure 2.8

With the previous Flux Weakening implementation we had problems of discontinuities in the passage between MTPA and flux weakening, we have resolved this problem considering also the resistance that before we neglected. If we consider the resistances we obtain the following system that describes the intersection between the current and the voltage limits.

$$\begin{cases} I_d^2 + I_q^2 = I_{lim}^2 \\ U_d^2 + U_q^2 = U_{lim}^2 \end{cases} \qquad (2.33)$$

Where

$$I_{lim}^2 = I_{d\_MTPA}^2 + I_{q\_MTPA}^2 \qquad (2.34)$$

But

$$\begin{cases} U_d = RI_d - \Omega_{me} L_q I_q \\ U_q = RI_q + \Omega_{me}(L_d I_d + \Lambda_{mg}) \end{cases} \qquad (2.35)$$

Well

$$\begin{cases} I_d = -\sqrt{I_{lim}^2 - I_q^2} \\ (RI_d - \Omega_{me} L_q I_q)^2 + [RI_q + \Omega_{me}(L_d I_d + \Lambda_{mg})]^2 = U_{lim}^2 \end{cases} \qquad (2.36)$$

Substituting *Id* in the second equation we have:

$$[-R\sqrt{I_{lim}^2 - I_q^2} - \Omega_{me} L_q I_q]^2 + [RI_q + \Omega_{me}(-L_d\sqrt{I_{lim}^2 - I_q^2} + \Lambda_{mg})]^2 = U_{lim}^2 \qquad (2.37)$$

Now we develop the equation considering like unknown the speed, in this way we obtain a second order equation

$$\left[L_q^2 I_q^2 + (-L_d\sqrt{I_{lim}^2 - I_q^2} + \Lambda_{mg})^2\right] * \Omega_{me}^2 + \left[2R\sqrt{I_{lim}^2 - I_q^2} L_q I_q - 2RI_q L_d\sqrt{I_{lim}^2 - I_q^2} + \Lambda_{mg}\right] * \Omega_{me} + R^2 I_{lim}^2 - U_{lim}^2 = 0 \qquad (2.38)$$

Well

$$A = L_q^2 I_q^2 + (-L_d\sqrt{I_{lim}^2 - I_q^2} + \Lambda_{mg})^2 \qquad (2.39)$$

$$B = 2R\sqrt{I_{lim}^2 - I_q^2} L_q I_q - 2RI_q L_d\sqrt{I_{lim}^2 - I_q^2} + \Lambda_{mg} \qquad (2.40)$$

$$C = R^2 I_{lim}^2 - U_{lim}^2 \qquad (2.41)$$

$$\Delta = B^2 - 4AC \qquad (2.42)$$

$$\Omega_{me12} = \frac{-B \pm \sqrt{\Delta}}{2A} \qquad (2.43)$$

But we are interested only at positive speeds so

$$\Omega_{me} = \frac{-B + \sqrt{\Delta}}{2A} \qquad (2.44)$$

Now that we have obtained an analytical expression of the speed in function of the q current, we can use again the polyfit to extract the *Iq* from the speed. In this case we have used a polyfit until the eleven order and we have built a matrix of coefficients K in the main matlab code, where every row of this matrix corresponds to the polyfit coefficients calculated with a particular current limit from the MTPA code. In the appendix is showed the code to do this, while here there are the plots of the curves of *speed(Iq)* (in red) and the *Iq(speed)* (in blu).

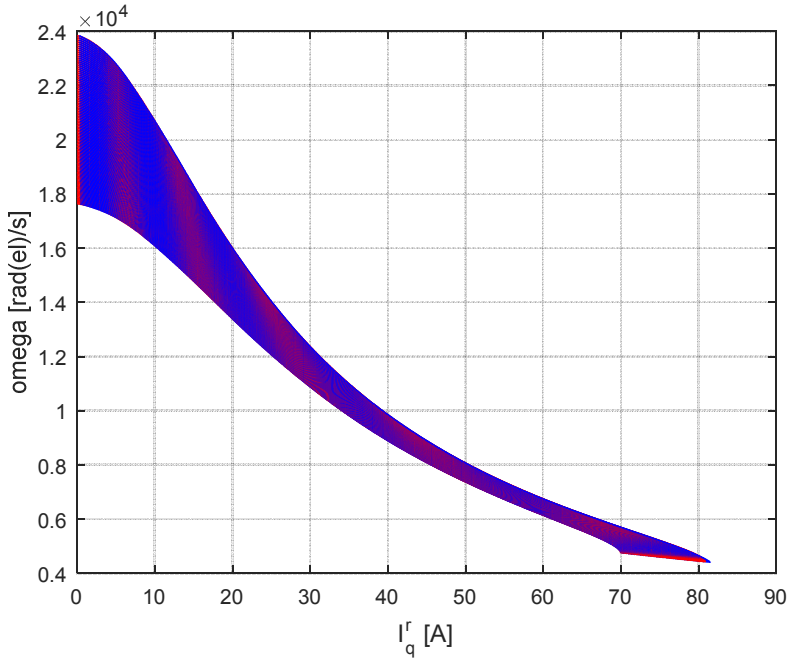


Figure 2.9

If we do the zoom in the lower right region that is our operating region we observe the errors of the approximation.

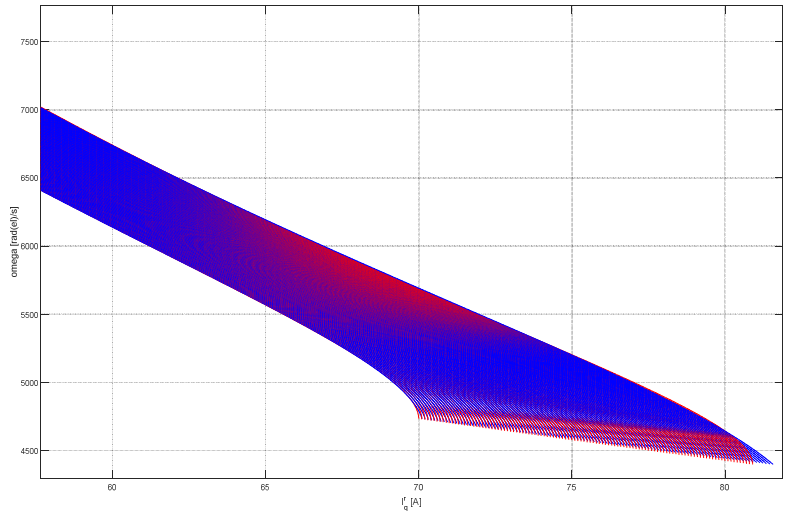


Figure 2.10

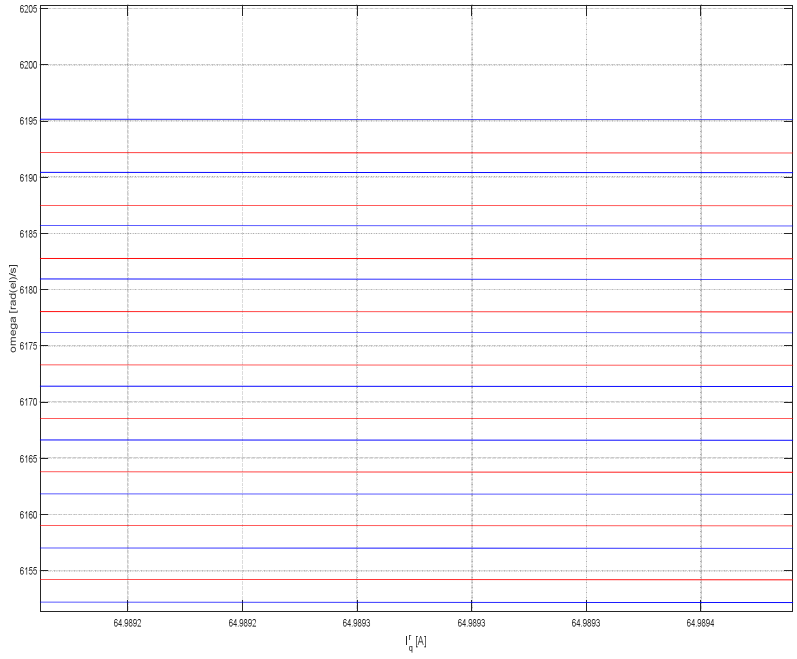


Figure 2.11

We can see that the error is about of 2 [rad el / sec] that results acceptable in our control and in this way we have resolved the problem of the discontinuity.

Once known the q-current is very easy to find the d-currents that is calculated like follows:

$$i_{d\,ref} = -\sqrt{|I|^2 - i_{q\,ref}^2} \qquad (2.45)$$

Where

$$|I| = \sqrt{i_{d\,MTPA}^2 + i_{q\,MTPA}^2} \qquad (2.46)$$

Finally we have obtained the references for our control, now let's see in detail the PI regulators utilized.

### *2.5 PI*

For the current regulators we have used simple PI with the proportional part and the integral part with the following function transfer.

$$R(s)_I = K_P + \frac{K_I}{s} \qquad (2.47)$$

With a regulator time constant equal to

$$\tau_R = \frac{K_P}{K_I} \qquad (2.48)$$

And the following Simulink scheme

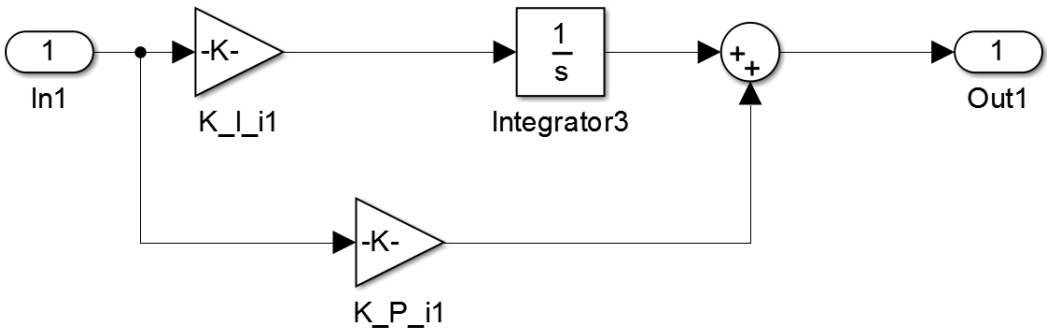


Figure 2.12

For the first calculation of the coefficients *KI* and *KP* we have written a code (that we can find in the appendix of this chapter) with particular specifications of cut frequency and phase margin, considering the machine like an SPM and doing the axis decoupling and the fem compensation. In this way we can consider the *d* loop and the *q* loop equals and we can calculate *KI* and *KP* once like follows. However, after the first calculation of *KP* and *KI,* we have to do a work of tuning to find the right coefficients watching the simulations with the full scheme of control. To simplify the work of discretization, we have written the PI with Matlab and we have added two typologies of anti-wind-up algorithm, one static and the other dynamic. Here there is the figure of the regulators.

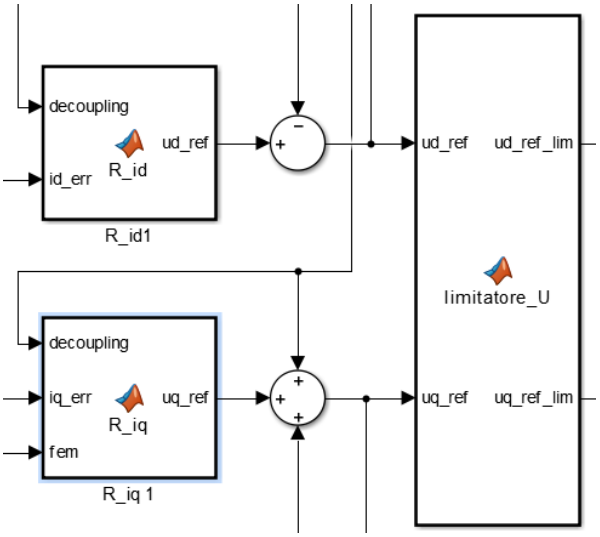


Figure 2.13

The anti-wind-up is a particular limiter of the integrator inside the regulator. It can be static or dynamic and we have implemented both choosing the dynamic and commenting the code part of the static one, but let's see what a dynamic anti wind-up does:

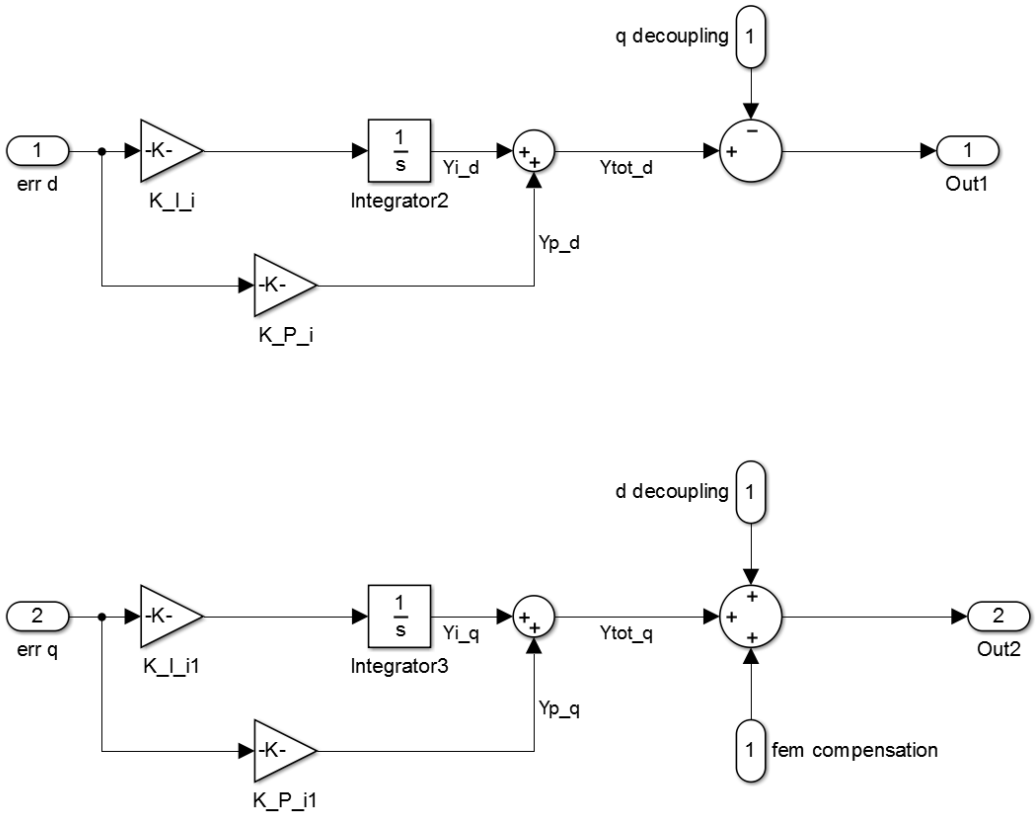


Figure 2.14

Like we can observe from the figure 2.14, in the d loop we have:

$$Y_{FF} = -q\ decoupling \qquad (2.49)$$

While in the q loop we have

$$Y_{FF} = d\ decoupling + fem\ compensation \qquad (2.50)$$

So we see that *Y_FF* can change and our algorithm can take it in to account because it does the followings operations:

We want respect this restriction

$$-U_{lim} \leq Y_I + Y_P + Y_{FF} \leq U_{lim} \qquad (2.51)$$

So, if we have

$$Y_I + Y_P + Y_{FF} \geq U_{\lim} \qquad (2.52)$$

Do this

$$Y_I = U_{lim} - Y_P - Y_{FF} \qquad (2.53)$$

While if we have

$$Y_I + Y_P + Y_{FF} \leq -U_{lim} \qquad (2.54)$$

Do this

$$Y_I = -U_{lim} - Y_P - Y_{FF} \qquad (2.55)$$

In this way our algorithm change the limits of the integrator when *Y_P + Y_FF* change and this is the motive therefore it is called dynamic anti-wind-up. Always in the appendix we can find the code used to do this.

### *2.6 Voltage limiter*

Done always in code, thanks this limiter, we are sure to remain in the linear zone of the inverter, so if we want to pass from the linear zone to the overmodulation zone and square wave (six steps control) we have to remove it. Are been implemented two types of it, one using Pitagora theorem and triangles similarity, the other using trigonometric theory. For the implementation in C-script is been selected the geometric one because it doesn't need of trigonometric form that are very heavy in terms of calculus for the DSP. Anyway in the appendix are reported both the codes.

### *2.7 Appendix*

a) Polyfit for MTPA

```
%%

iq=-150:150;

%    dopo diverse pagine di conti, ho trovato la seguente equazione :
%    -4 *m^2 +6p*flux*iq *m +9p^2(Ld-Lq)^2iq^4 = 0
%    risolvendo l'equazione di secondo grado, si arriva all'espressione
%    della coppia come segue

m=3/4*p*iq.*(flux_mg+sqrt((flux_mg)^2+4*(Ld-Lq)^2*iq.^2));

%%

%    esprimiamo adesso la iq in funzione di m ( che e' proprio quello che
%    ci interessa ) e lo si fa con una polyfit

Cq=polyfit(m,iq,8);
iq=(Cq(1)*m.^8 + Cq(2)*m.^7 + Cq(3)*m.^6 + Cq(4)*m.^5 + Cq(5)*m.^4 +
Cq(6)*m.^3 + Cq(7)*m.^2 + Cq(8)*m + Cq(9));

Cq;     % coefficienti della polinomiale
```

b) Polyfit for FW

```
for i=1:110

    I_lim_MTPA=70+i/10-0.1;
    Iq = linspace(0,I_lim_MTPA,100000);
    A = ((Lq.*Iq).^2 + Ld^2*(I_lim_MTPA^2-Iq.^2) + flux_mg^2 -
         2*Ld*flux_mg*sqrt(I_lim_MTPA^2-Iq.^2));
    B = (2*R.*Iq.*(sqrt(I_lim_MTPA^2-Iq.^2).*(Lq-Ld) + flux_mg));
    C = (R*I_lim_MTPA)^2 - U_lim^2;
    Delta = B.^2 - 4.*A.*C;
    omega = (-B + sqrt(Delta))./(2.*A);

     Kq = polyfit(omega,Iq,10);

     Iq = (Kq(1)*omega.^10 + Kq(2)*omega.^9 + Kq(3)*omega.^8 +
           Kq(4)*omega.^7 + Kq(5)*omega.^6 + Kq(6)*omega.^5 +
           Kq(7)*omega.^4 + Kq(8)*omega.^3 + Kq(9)*omega.^2 +
           Kq(10)*omega.^1 + Kq(11));

     Kq(i,:)=Kq;
     I_blocco_FW(:,i)=I_lim_MTPA(1,:);

     KQ(i,:)=Kq(1,:);
end

I_blocco_FW
KQ
```

c) Ki and Kp calculation

```
%%%% calcolo dei K_I e K_P del regolatore PI degli anelli di corrente

%% dati

Fs=16e3;
Ts=1/Fs;
R=0.0951;
L=211e-6;

%% specifiche

f_a=Fs/50;                          %frequenza di attraversamento [Hz]
omega_a=2*pi*f_a;                   %velocità di attraversamento  [rad/s]
margine_di_fase_richiesto=60*(pi/180);     %margine di fase in radianti

%% funzione di trasferimento GH per gli anelli di corrente
 % GH= K_I * (1+s*tr)/s * 1/(1+s*(3/2)*Ts) * 1/(R+s*L)

%% calcolo K_I e K_P

% margine_di_fase_richiesto= pi +
% + [arctg(omega_a*tr)-(pi/2)-arctg(omega_a*3/2*Ts)-arctg(omega_a*L/R)]
% trovo l'incognita tr

tr=(1/omega_a)*tan(margine_di_fase_richiesto-
pi+(pi/2)+atan(omega_a*3/2*Ts)+atan(omega_a*L/R));

% impongo |GH(jomega_a)|=1 con il valore di tr appena trovato e trovo K_I

K_I=R*sqrt(((omega_a)^2*(1+(3/2*Ts*omega_a)^2)*(1+(omega_a*L/R)^2))/(1+(ome
ga_a*tr)^2));

% calcolo K_P

K_P=K_I*tr;

K_I
K_P
```

d) PI discretized with static and dynamic anti-wind-up

```
function ud_ref = R_id( U_lim, decoupling, id_err)

%%%% controllore di corrente fatto in codice perche' va meglio poi per la
%%%% discretizzazione

%% dati

K_I=470.8638*0.3;
K_P=0.3754*1.7;

%% algorithm for ud_ref

persistent ud_int
```

```
if ( isempty(ud_int) )
    ud_int=0;
end

% discretizzazione di Eulero

ud_int=ud_int + id_err*Ts;

% anti wind-up dinamico

if K_I*ud_int+K_P*id_err-decoupling>U_lim
   ud_int=(U_lim-K_P*id_err+decoupling)/K_I;
end

if K_I*ud_int+K_P*id_err-decoupling<-U_lim
   ud_int=(-U_lim+decoupling-K_P*id_err)/K_I;
end

% cosa fa il PI

ud_ref=K_P*id_err+K_I*ud_int;

%% anti wind-up statico
%
% if ud_ref >= U_lim
%     ud_int=(U_lim-K_P*id_err)/K_I;
%     id_ref=U_lim;
% end
%
% if ud_ref <= -U_lim
%     ud_int=(-U_lim-K_P*id_err)/K_I;
%     ud_ref=-U_lim;
% end
```

e) Voltage limiter

```
function [ud_ref_lim, uq_ref_lim] = limitatore_U(ud_ref, uq_ref, U_lim)

%%%%    calcolo del modulo (abs voltage value calculation)
%% geometric method
% k=modulo_U/U_lim;
%
%
% if k>1
%     ud_ref_lim=ud_ref/k;
%     uq_ref_lim=uq_ref/k;
% else
%     ud_ref_lim=ud_ref;
%     uq_ref_lim=uq_ref;
% end
%% trigonometric method
modulo_U=sqrt(abs(ud_ref)^2+abs(uq_ref)^2);
fase_U=pi+atan(uq_ref/ud_ref);
if modulo_U > U_lim
   modulo_U=U_lim;
end
ud_ref_lim=modulo_U*cos(fase_U);
uq_ref_lim=modulo_U*sin(fase_U);
```

# Chapter 3:

# The switching frequency

### *3.1 Introduction*

It's necessary to discuss about the switching frequency and the modulation frequency index. Considering the fact that is simple filtering the high frequency harmonics, it is better to choose a switching frequency very high with the only disadvantage of the switching losses in the inverter. We have a good relation between the carrier signal and the control one, if the modulation frequency index $m_f$ is high. The limit therefore $m_f$ is considered big or small is $m_f = 21$.

$\boldsymbol{m_f} > 21$

The subharmonic amplitudes caused by an asynchronous PWM modulation are small for high values of $m_f$. So, we can hold constant the carrier frequency while the frequency of the control signal is changing, and we have non-integer values of $m_f$, like we can understand by the following formula.

$$m_f = \frac{f_{switching}}{f_{motor}} \qquad (3.1)$$

$\boldsymbol{m_f} < 21$

For small values of $m_f$, the carrier and the control signal have to be synchronous. In order to reach this, the modulation frequency index $m_f$ has to be integer, if $m_f$ is not an integer number we subharmonics of the fundamental frequency that are unwanted. In this way the carrier has to change the frequency with the frequency of the control signal. In particular $m_f$ has to be also odd to have odd and half-wave symmetry.

### *3.2 Variable switching frequency*

We have seen that when the speed of the motor increases, the modulation index of the frequency *mf* is very low, in particular when we are at full speed, the *mf* is around 12 and we can see clearly the subharmonics in the system. The theory said us that when the modulation index of the frequency is smaller than 21, it is better to work with the maximum odd integer value of *mf*. In this way we should see an improvement in the harmonic spectrum, so we have to write an algorithm that can change the switching frequency in a dynamic way to prove the theory.

The maximum switching frequency of the inverter is 16 [KHz], so until the mf is smaller than 21 we can work with a constant switching frequency of 16[KHz], that is the best solution in terms of harmonics because we can modulate better (more times) our signal. The first step is to write an algorithm that calculate the modulation index *mf* and when this is smaller than 21, it finds the exact switching frequency that gives us the value of *mf* reached. The maximum current value of *mf* is calculated like follows:

$$f_{switching\_max} = 16000\ [Hz] \qquad (3.2)$$

$$f_{motor} = \frac{\Omega_{me}}{2\pi} \qquad (3.3)$$

Because *Ω_me* is in [rad el/s].

$$m_{f\_max} = \frac{f_{switching\_max}}{f_{motor}} \qquad (3.4)$$

So if *mf* is bigger that 21 we have:

$$f_{switching} = f_{switching\_max} \qquad (3.5)$$

$$m_f = m_{f\_max} \qquad (3.6)$$

For the case *mf* smaller than 21, we have written two algorithms: one that finds the switching frequency with the max integer odd value of *mf*, and the other one which finds the switching frequency necessary to have the max integer odd and multiple of 3 value of *mf* and they are been reported in the appendix of this chapter.

Let's start with the first one:

When the value of *mf* is smaller than 21, we calculate the max integer closer to the current *mf* with the Matlab command *ceil.* Now we need that this integer is odd, so we divide the integer by two and we calculate the closer integer of it thanks the function *round* in Matlab. Well if the last two operations give the same number, it means that the previous number (the one calculated with the Matlab function *ceil*) was even and we need to sum 1, otherwise it means that the previous number was odd and so we can take it directly. Finally we have found the *mf* which we wanted and now we can calculate the switching frequency that corresponds to it with the inverse of the formula 3.1 like follows:

$$f_{switching} = m_f * f_{motor} \qquad (3.7)$$

The block in Simulink has *Ω_me* in input and f_switching in output calculated with the algorithm just described.

The second algorithm that finds the switching frequency which gives an integer multiple of 3 and odd value of *mf*, it is composed only by two if cycles very easy to understand only watching the code in the appendix.

The second step consists to find a way to convey at my carriers the variable switching frequency just calculated. We did it analogically using an electrical circuit in Plecs that we can observe in the following figure.

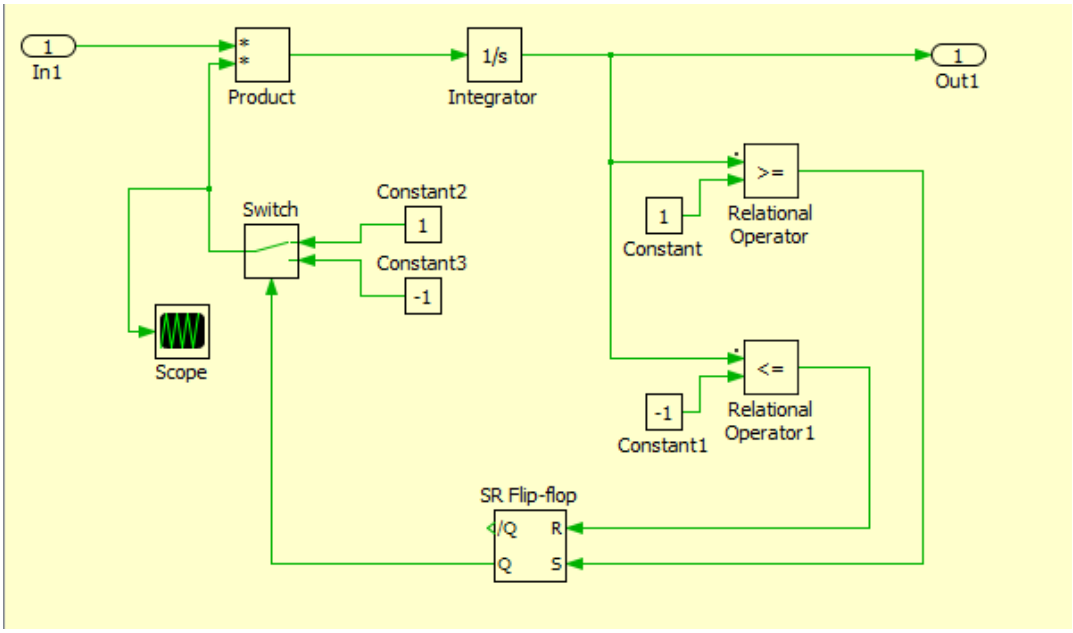


Figure 3.1

The input is a constant C multiplied by 1 or -1 and later integrated, in fact the carrier is the integral of a square wave. The switch that controls the sign of the constant is controlled by a Flip-flop SR that orders the switch to limit the carrier between 1 and -1. Now we have to find the relation between the ''constant'' C and the variable switching frequency. It is a simple geometric relation and it is easy to understand watching the figure below of a triangular wave.

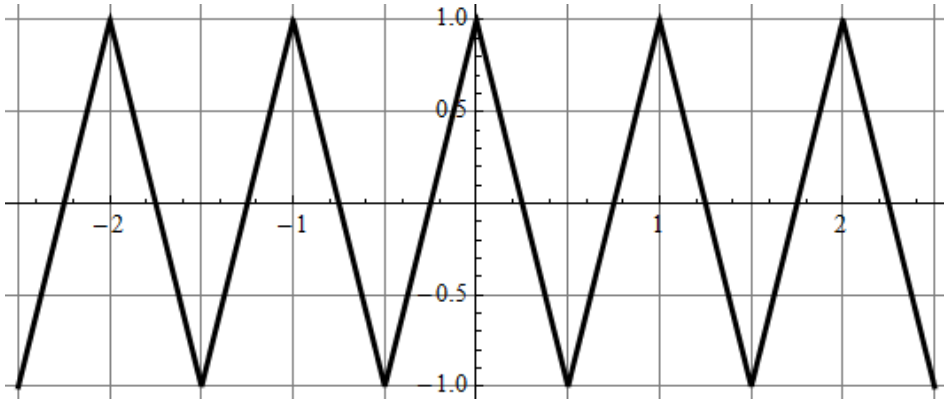


Figure 3.2

The ''constant'' C is the max value of the square wave and it is equal to the rise inclination of the carrier.

Let's consider the triangle of the figure 3.3.

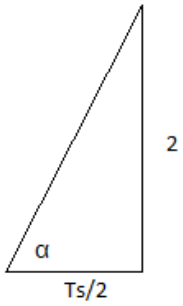


Figure 3.3

Well we can write:

$$C = tg(\alpha) = \frac{2}{\frac{T_s}{2}} = \frac{4}{T_s} = 4f_{switching} \qquad (3.8)$$

$$C = 4f_{switching} \qquad (3.9)$$

The 3.9 is the final relation reached between the switching frequency and the input for the electrical circuit that makes the carrier.

The final scheme used in the simulation to have a variable switching frequency is the following.

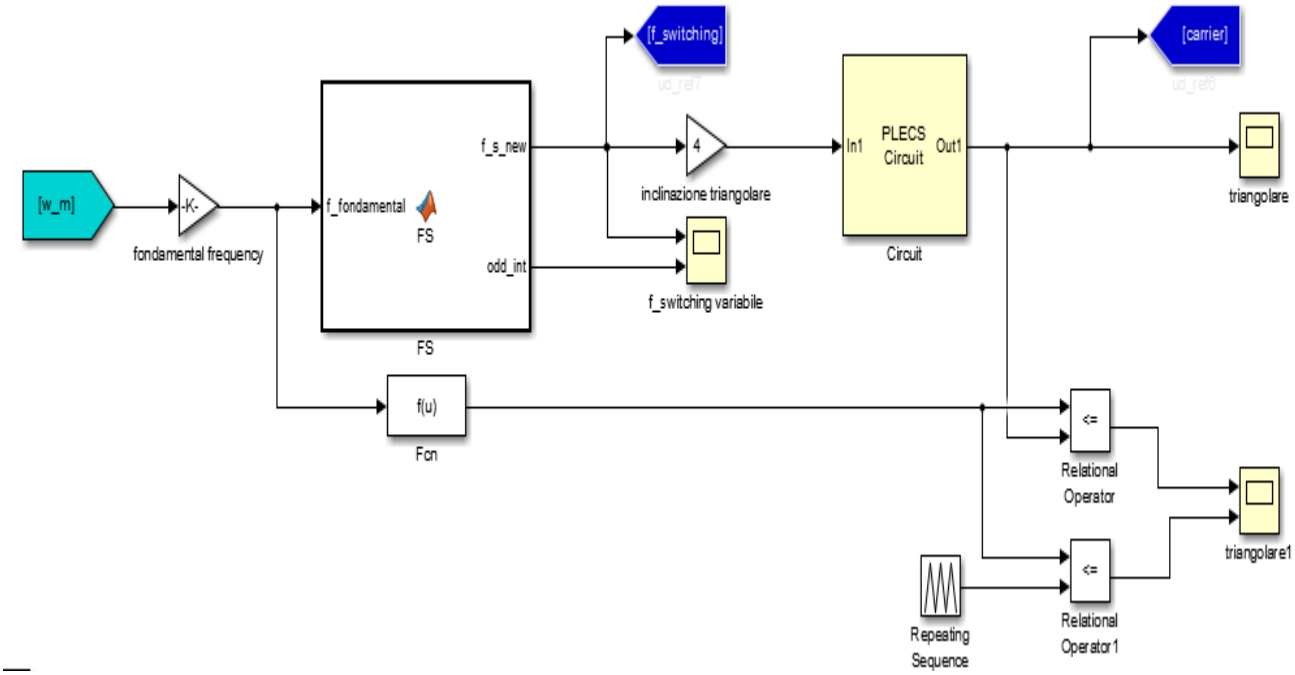


Figure 3.4

### *3.3 Best solution*

To verify what algorithm is better for our control, we observe the harmonic spectrum of the modulation of three sinusoidal signals modulated with the fixed frequency at 16 [KHz] and with the variable switching frequency just described. The period where we want to calculate the spectrum is the period of the fundamental of the motor at full speed.

$$full\ speed = 19000 \quad [rpm] \qquad (3.10)$$

$$\Omega_{motor} = 19000 * 4 * \frac{2\pi}{60} \quad \left[\frac{rad_{el}}{sec}\right] \qquad (3.11)$$

$$f_{motor} = \frac{\Omega_{motor}}{2\pi} \quad [Hz] \qquad (3.12)$$

$$T_{f_{motor}} = \frac{1}{f_{motor}} \quad [s] \qquad (3.14)$$

We want to compare the three different spectrums in the case of the phase signals and in the case of the line to line signals both. To convert the phase signals in the line to line we have used the simple trick in the figure 3.5.

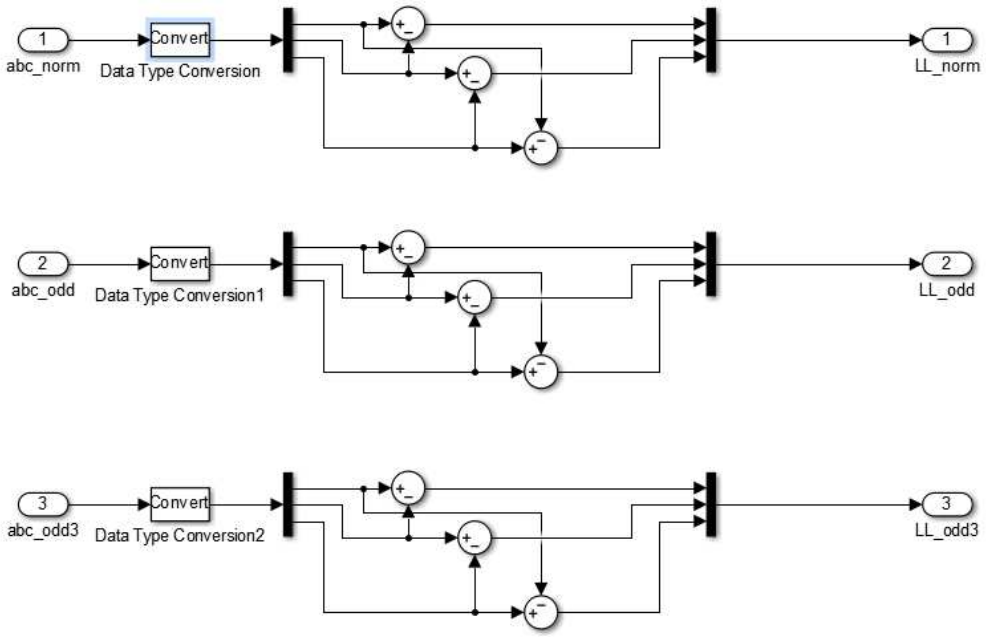


Figure 3.5

Where the inputs are the three different modulations of the phase signals and the outputs are the line to line modulations.

The scheme to verify what is the best algorithm is the one described in the figure 3.6.

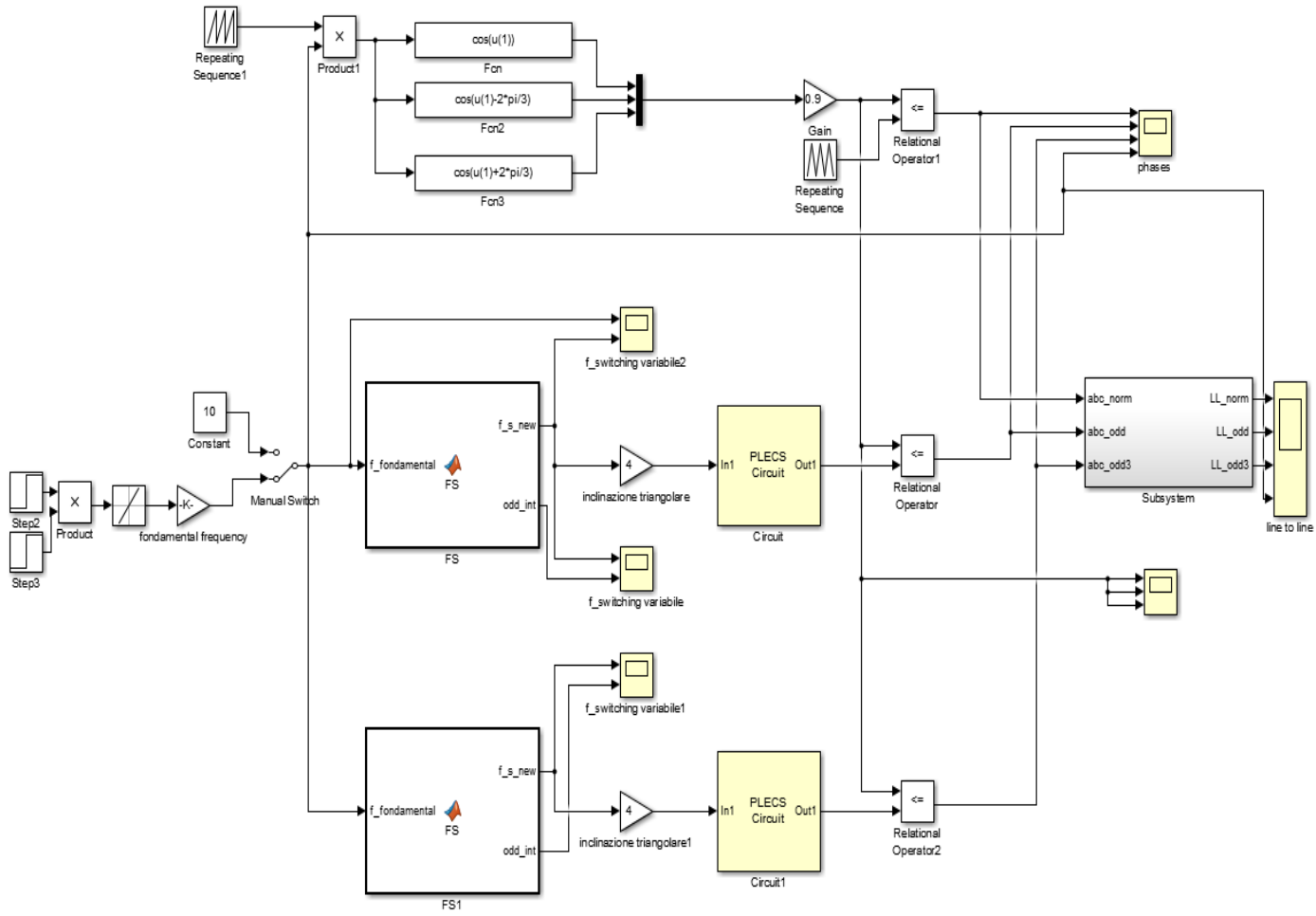


Figure 3.6

Now let's see the results.

The variable switching frequency calculated with the first algorithm and the fundamental frequency of the motor from zero to the full speed following the requirements of the project are presented in the figure 3.7

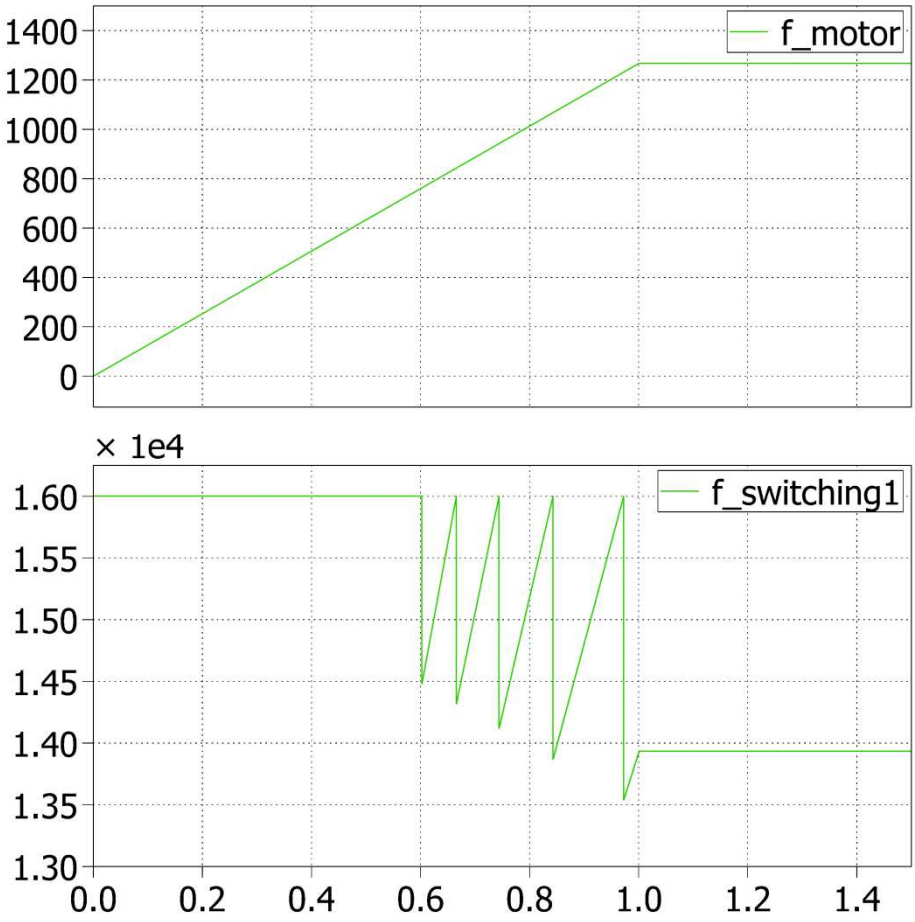


Figure 3.7

The variable switching frequency calculated with the first algorithm and the modulation frequency index $m_f$ are showed in figure 3.8.

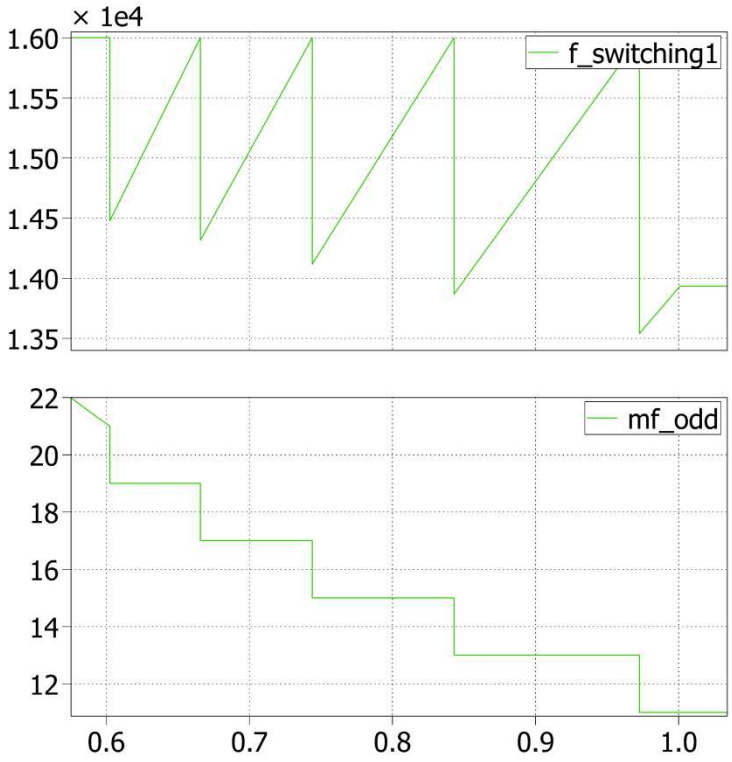


Figure 3.8

The variable switching frequency calculated with the second algorithm and the corresponding modulation frequency index $m_f$ are showed in figure 3.9.

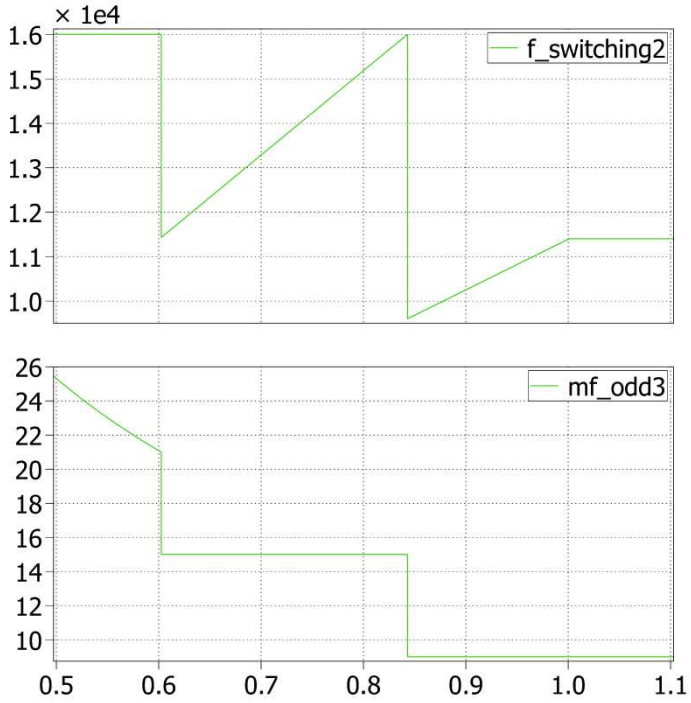


Figure 3.9

From now in the figures, the first image of the figure will represent the modulation at constant frequency, the second the modulation done with variable switching calculated using the first algorithm and the third the other. Let's consider a period at full speed of the three modulations of the phase signals.

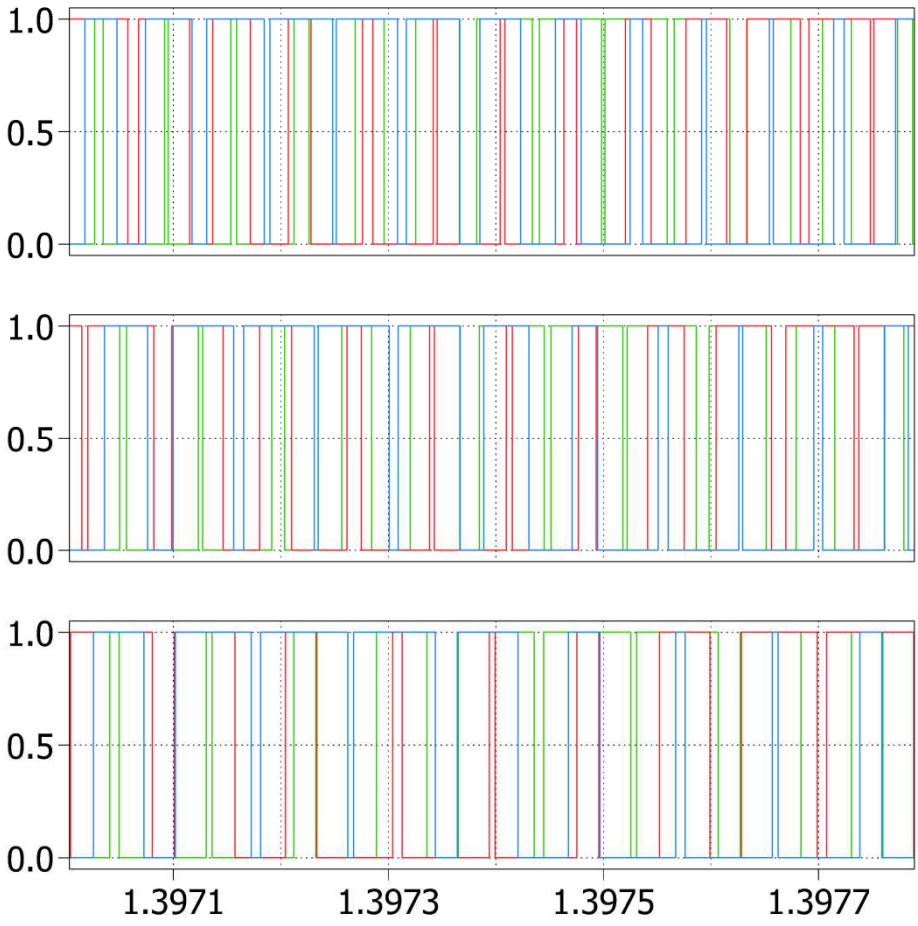


Figure 3.10

In the figures 3.11 and 3.12 we can observe the FFT (Fast Fourier Transform) done until the 50$^{th}$ harmonic and 18$^{th}$ respectively.

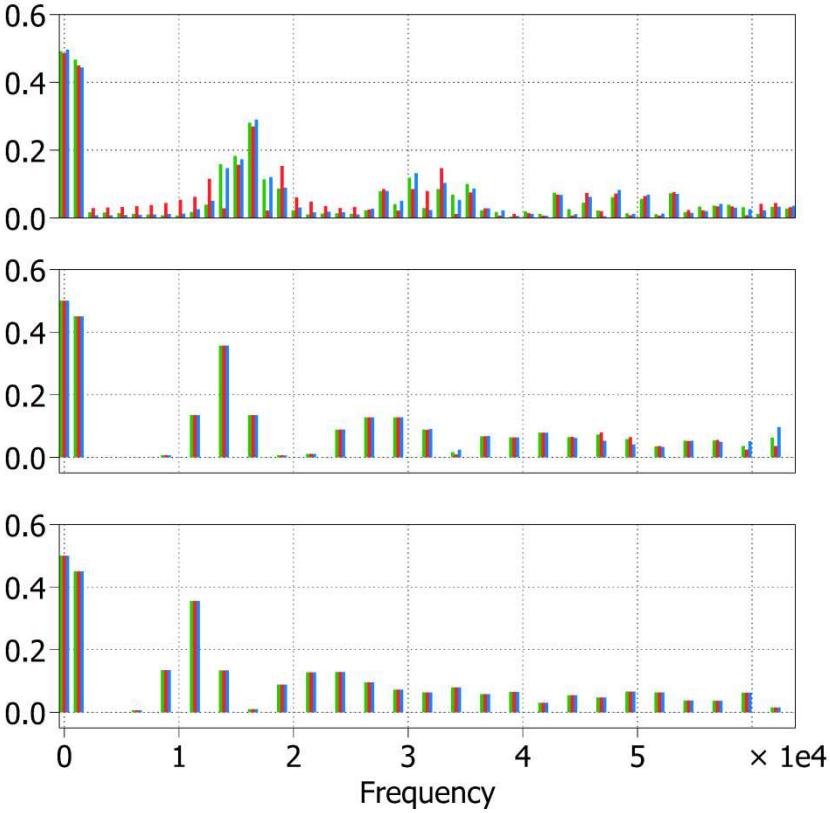


Figure 3.11

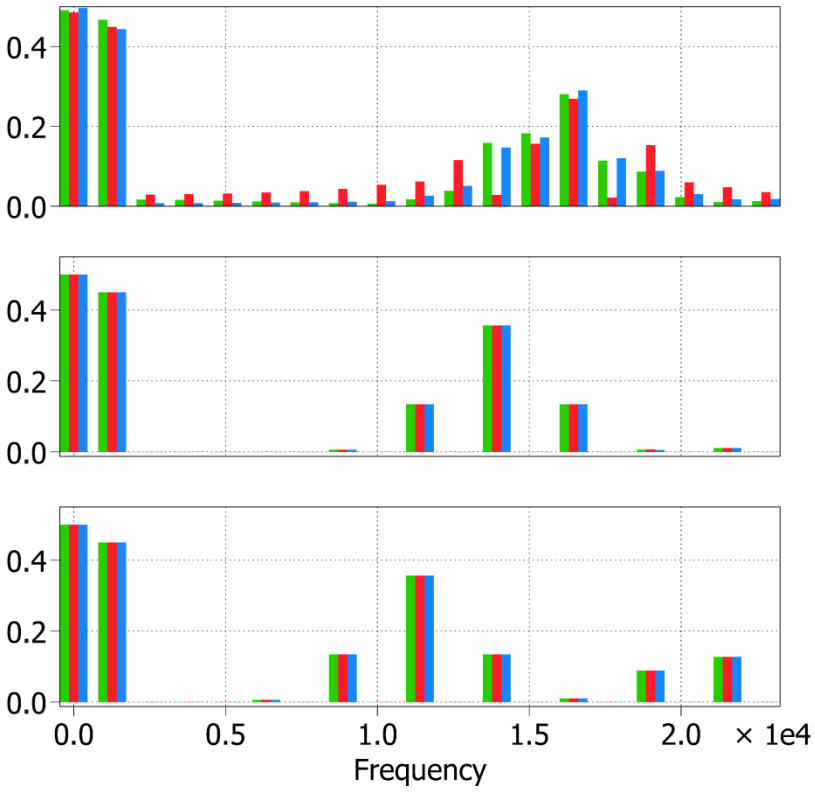


Figure 3.12

But what is really interesting for us is the behaviour of the modulations of the line to line signals, well it is proposed below.

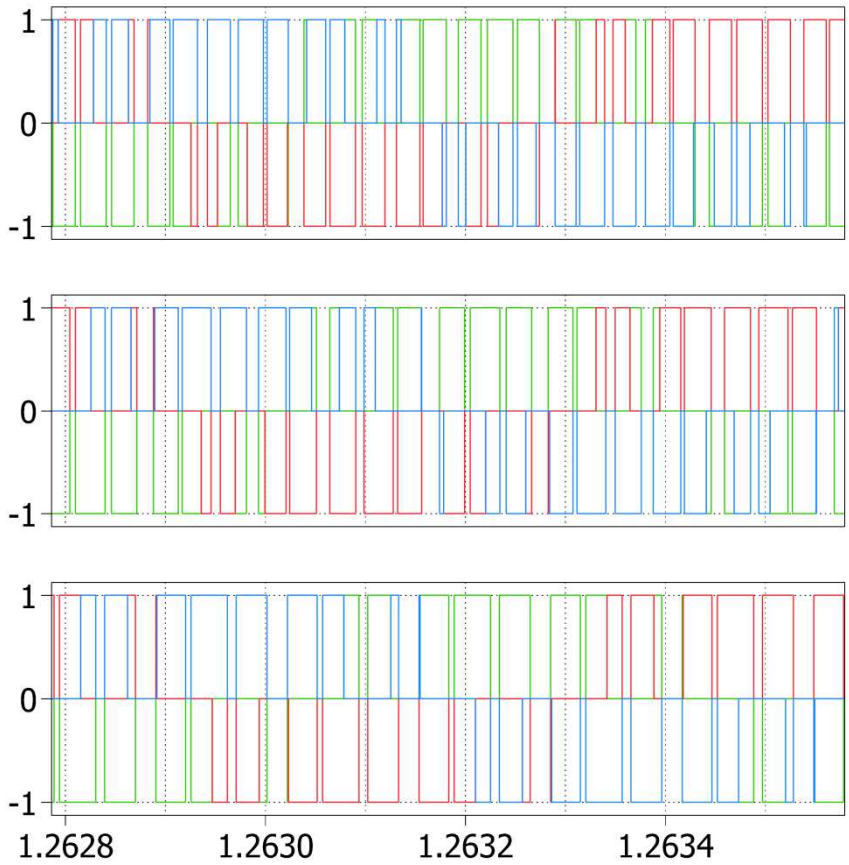


Figure 3.13

We can observe that there isn't a continue component and in the figures 3.14 and 3.15 we can observe the FFT (Fast Fourier Transform) until the $50^{th}$ harmonic and $20^{th}$ respectively.

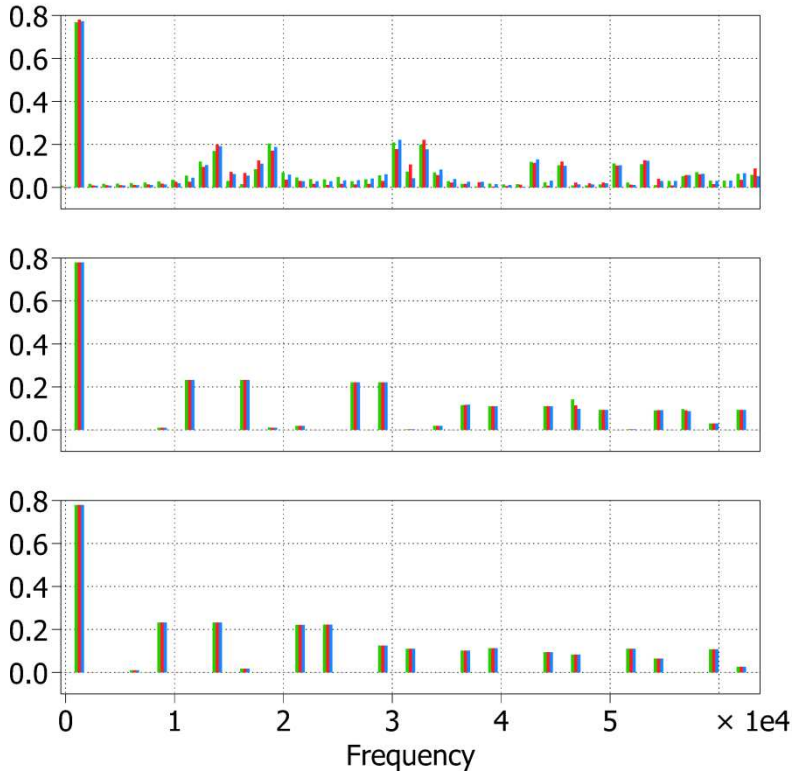


Figure 3.14

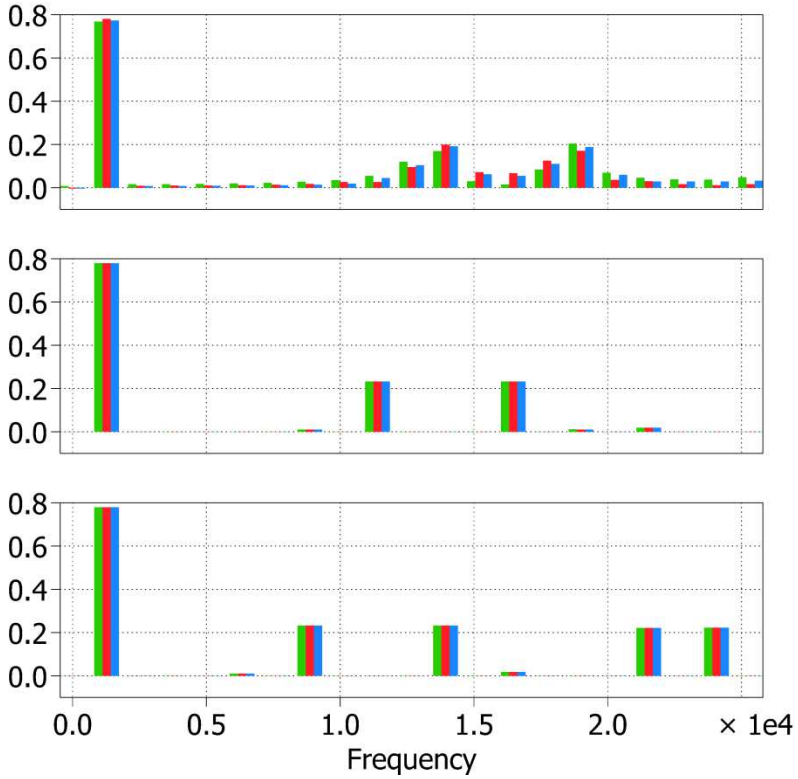


Figure 3.15

We can see clearly an improvement, in terms of harmonics, using the modulations with variable switching. In the last figure, we can see that the modulation done with an *mf* integer and odd presents a ninth harmonic and a little bit of seventh, but with the line to line signals, we didn't expect harmonics of order multiple of three, so it is the harmonic caused by the switching. Instead the modulation done with a *mf* integer, odd, and multiple of three, presents the seventh and a little bit of fifth, so finally we will implement the first algorithm to control the switching frequency of our inverter.

During the simulations done, the discontinuities of the switching frequency of figure 3.7 were damaging for the control, so we have decided finally to discretize all at the final switching frequency founded using the first algorithm that is equal to 13933.3 [Hz] and it guaranties a frequency modulation index at full speed equals to 11.

### *3.4 Appendix*

First algorithm and second (commented)

```
function [f_s_new,odd_int] = FS(f_fondamental)

%%%% blocco che mi calcola la massima frequenza di switching tale per cui
%%%% quando l'indice di modulazione mf e' minore di 21, esso sia il massimo
intero dispari


f_s_max = 16000;% hz frequenza massima di switching

f_s_new = f_s_max;

m_f_max = f_s_max/f_fondamental;   % massimo indice di modulazione

m_f = m_f_max;

m_f_new = m_f;

if m_f >= 21
    f_s_new = f_s_max;
else

    m_f_new = ceil(m_f);

    m_f_pari = m_f_new/2;                    % mi serve per trovare mf
dispari con  un  trucchetto

    if round(m_f_pari) == m_f_pari
        m_f_new_dispari = m_f_new + 1;
    else
        m_f_new_dispari = m_f_new;
    end

    m_f_new = m_f_new_dispari-2;
    f_s_new = (m_f_new_dispari-2) * f_fondamental;

% % voglio che mf sia intero, dispari, e multiiplo di 3
%
%       if  m_f < 21  &&  m_f >= 15
%           m_f_new = 15;
%           f_s_new = m_f_new * f_fondamental;
%       end
%
%       if  m_f < 15
%           m_f_new = 9;
%           f_s_new = m_f_new * f_fondamental;
%       end
end

odd_int = m_f_new;
```

# Chapter 4:

# Overmodulation, square wave and selective harmonics cancellation

### *4.1 Introduction*

In this chapter we will analyse the behaviour of our motor controlling the inverter in overmodulation and successively in square wave. We have reached for some algorithms in the platform IEEE and we have found a very good article of the professors Silverio Bolognani and Mauro Zigotto entitled '' Novel digital continuous control of SVM inverters in the Overmodulation range''[1]. In this article is explained a smart way to do the transition of the PWM control from the linear zone to the overmodulation and the square wave. Using the modulation in square wave, the harmonics at low frequency make a lot of problems so we have proposed a selective harmonic cancellation algorithm always described in this chapter.

### *4.2 Overmodulation*

The first thing to do to apply the overmodulation is removing the voltage limiter so that the regulators output can surpass the voltage limit if necessary. We define the modulation index like follows:

$$M = \frac{U_1}{\frac{2U_{DC}}{\pi}} = \frac{\pi U_1}{2U_{DC}} \qquad (4.1)$$

where U_1 is the fundamental amplitude and U_DC is the DC-link voltage of our inverter. Now we consider the space vector of the voltages using the Clarke Transform.

$$\begin{cases} u_a(\theta) \\ u_b(\theta) \\ u_c(\theta) \end{cases} \qquad (4.2)$$

We apply the Clarke Transform definition

$$\boldsymbol{u}(\theta) \triangleq \frac{2}{3}\left[u_a(\theta) + u_b(\theta)e^{\frac{j2\pi}{3}} + u_c(\theta)e^{\frac{j4\pi}{3}}\right] \qquad (4.3)$$

Now we consider a space vector of amplitude *r* and phase $\theta_1$ in a certain instant. When it turns and it remains inside the hexagon like we can see in the figure, we are sure to be in the linear zone of PWM, but when the amplitude of the space vector becomes bigger than the maximum radius of the circle inscribed in the hexagon, we are outside of the linear zone and we have to modify its trajectory. The maximum radius therefore we are outside of the linear zone is the following:

$$r_{max} = U_1 = \frac{U_{DC}}{\sqrt{3}} \qquad (4.4)$$

Then if

$$M > \frac{\pi \frac{U_{DC}}{\sqrt{3}}}{2U_{DC}} = 0.907 \qquad (4.5)$$

We have to apply our algorithm.

The space vector $\boldsymbol{u}$ is well represented from 0 to $\theta_1 = \alpha_g$ like we can observe in the following figure.

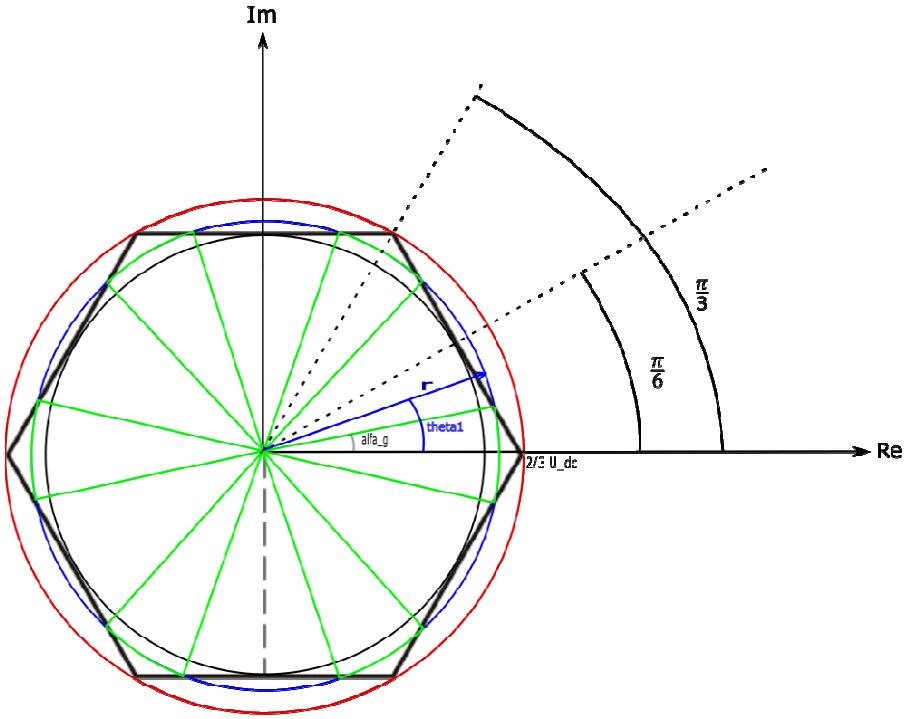


Figure 4.1

We have a space vector of reference that is turning and it is the blue one, then we have the trajectory of the space vector reached that is the green one. So the algorithm that we have implemented does the followings things:

Considering the first sector from 0 to $\frac{\pi}{3}$, until the reference space vector is inside the hexagon, the green space vector is equal the blue one, so from $\theta_1 = 0$ to $\theta_1 = \alpha_g$ we have

$$\boldsymbol{u} = \boldsymbol{u_{ref}} \qquad (4.6)$$

From $\theta_1 = \alpha_g$ to $\theta_1 = \frac{\pi}{6}$ we have that the reference space vector is outside the hexagon so we fix the green space vector at the phase $\theta_1 = \alpha_g$ so we have

$$\boldsymbol{u} = \left|\boldsymbol{u_{ref}}\right| e^{j\alpha_g} \qquad (4.7)$$

From $\theta_1 = \frac{\pi}{6}$ to $\theta_1 = \frac{\pi}{3} - \alpha_g$ we have that the reference space vector is still outside from the hexagon, but now we fix our green space vector at the phase $\theta_1 = \frac{\pi}{3} - \alpha_g$ so we have

$$\boldsymbol{u} = \left|\boldsymbol{u_{ref}}\right| e^{j(\frac{\pi}{3} - \alpha_g)} \qquad (4.8)$$

From $\theta_1 = \frac{\pi}{3} - \alpha_g$ to $\theta_1 = \frac{\pi}{3}$ the reference space vector is returned inside the hexagon, so the green one corresponds exactly with the blue one and we have again

$$\boldsymbol{u} = \boldsymbol{u_{ref}} \qquad (4.9)$$

For the others five sectors of the hexagon, we repeat the same things in a symmetric way and we obtain finally the upper figure 4.1.

The green space vector that goes out from our algorithm has the following expression for the first sector

$$\boldsymbol{u} = |\boldsymbol{u}| e^{j\theta} \qquad (4.10)$$

With

$$\theta = \begin{cases} \theta_1 & for \;\; 0 \leq \theta_1 \leq \alpha_g \;\; and \;\; \frac{\pi}{3} - \alpha_g \leq \theta_1 \leq \frac{\pi}{3} \\ \alpha_g & for \;\; \alpha_g \leq \theta_1 \leq \frac{\pi}{6} \\ \frac{\pi}{3} - \alpha_g & for \; \frac{\pi}{6} \leq \theta_1 \leq \frac{\pi}{3} - \alpha_g \end{cases} \qquad (4.11)$$

With this algorithm we can do a continuous transition from the linear zone to overmodulation and the six step, in fact in the bottom there are the condition therefore we are in square wave:

$$M = 1 \qquad (4.12)$$

$$U_1 = \frac{2}{3} U_{DC} \qquad (4.13)$$

$$\alpha_g = 0 \qquad (4.14)$$

The first thing to do to implement the algorithm is finding an expression of the angle $\alpha_g$ in function of $|\boldsymbol{u}|$ and the voltage of the DC-link.

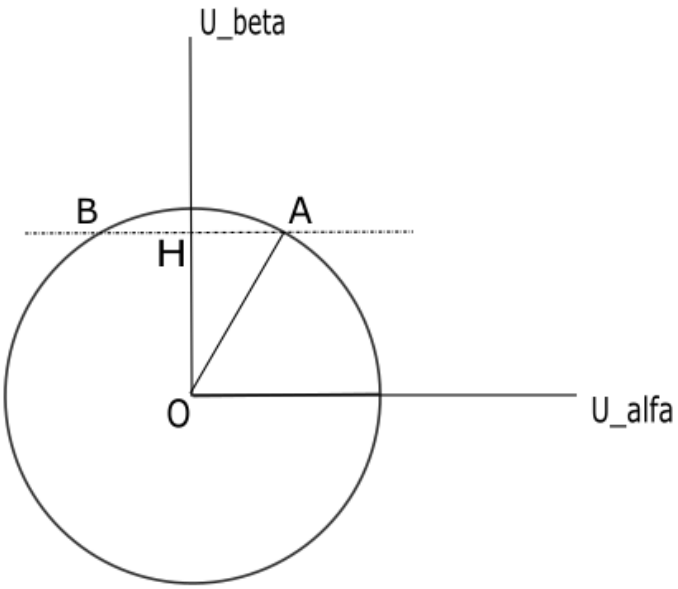


Figure 4.2

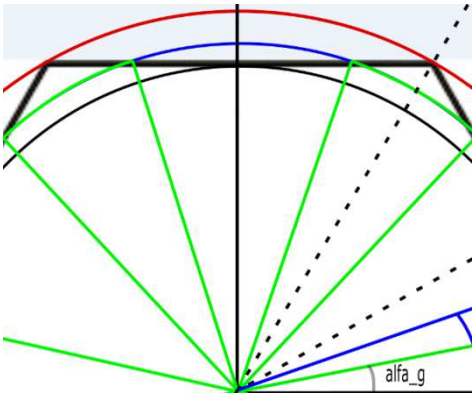


Figure 4.3

AB is the intersection between the side of the hexagon and the circle made from the reference space vector, OH is the apotema of the hexagon and the angle $H\hat{O}A$ is equal to $\frac{\pi}{2} - \frac{\pi}{3} - \alpha_g$.

$$AB = \frac{2}{3} U_{DC-link} \qquad (4.15)$$

$$OH = \frac{\sqrt{3}}{2} AB \qquad (4.16)$$

So the space vector OA has the followings coordinates:

$$\begin{cases} u_\alpha = \sqrt{|\boldsymbol{u}|^2 - {u_\beta}^2} \\ u_\beta = OH = \dfrac{\sqrt{3}}{2} AB = \dfrac{\sqrt{3}}{2}\dfrac{2}{3} U_{DC-link} = \dfrac{U_{DC-link}}{\sqrt{3}} \end{cases} \qquad (4.17)$$

Now let's write the last system 4.17 in polar coordinates

$$\begin{cases} r = |\boldsymbol{u}| \\ r\sin(\theta) = OH = \dfrac{U_{DC-link}}{\sqrt{3}} \end{cases} \qquad (4.18)$$

Then

$$\sin\theta = \frac{1}{\sqrt{3}} \frac{U_{DC-link}}{|\boldsymbol{u}|} \qquad (4.19)$$

Therefore

$$\theta = \arcsin\left(\frac{1}{\sqrt{3}} \frac{U_{DC-link}}{|\boldsymbol{u}|}\right) \qquad (4.20)$$

But

$$\theta = \frac{\pi}{2} - H\hat{O}A = \frac{\pi}{3} + \alpha_g \qquad (4.21)$$

So finally we have

$$\alpha_g = \theta - \frac{\pi}{3} = \arcsin\left(\frac{1}{\sqrt{3}} \frac{U_{DC-link}}{|\boldsymbol{u}|}\right) - \frac{\pi}{3} \qquad (4.22)$$

In the figure below is showed the Simulink block that do the algorithm and in the appendix of this chapter is reported the full code.

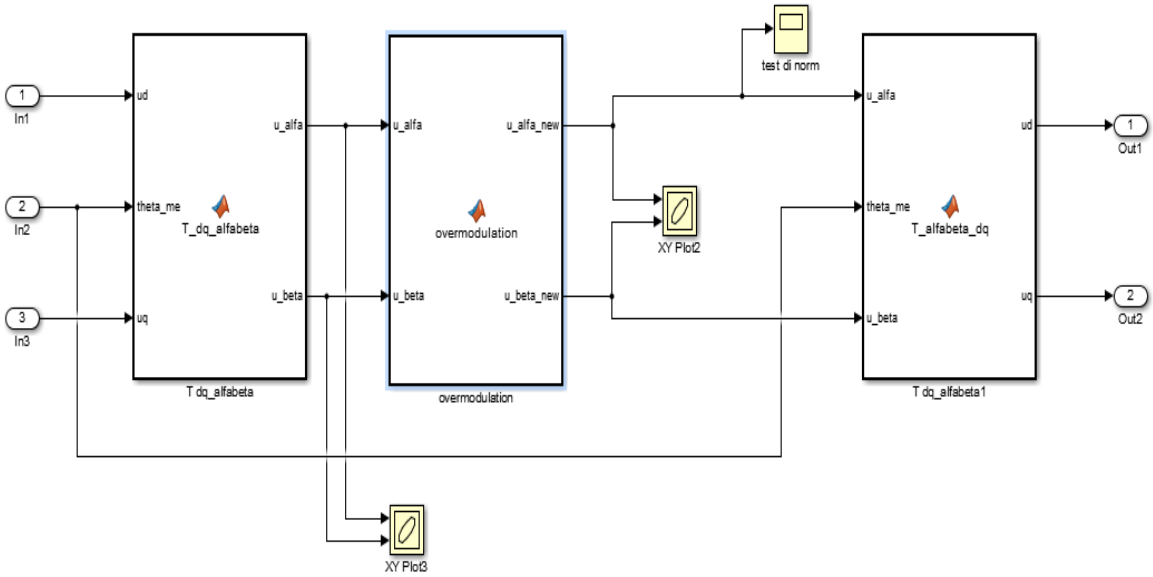


Figure 4.4

We can see that we need of two transformations because the algorithm works in alfa-beta, while our control is in a rotating reference system d-q. Now we want to show how the wave shapes of the voltages alfa-beta change during the simulation, from the linear zone of PWM to the overmodulation and square wave. In the followings figures are showed the alfa-beta voltages in the previous order.

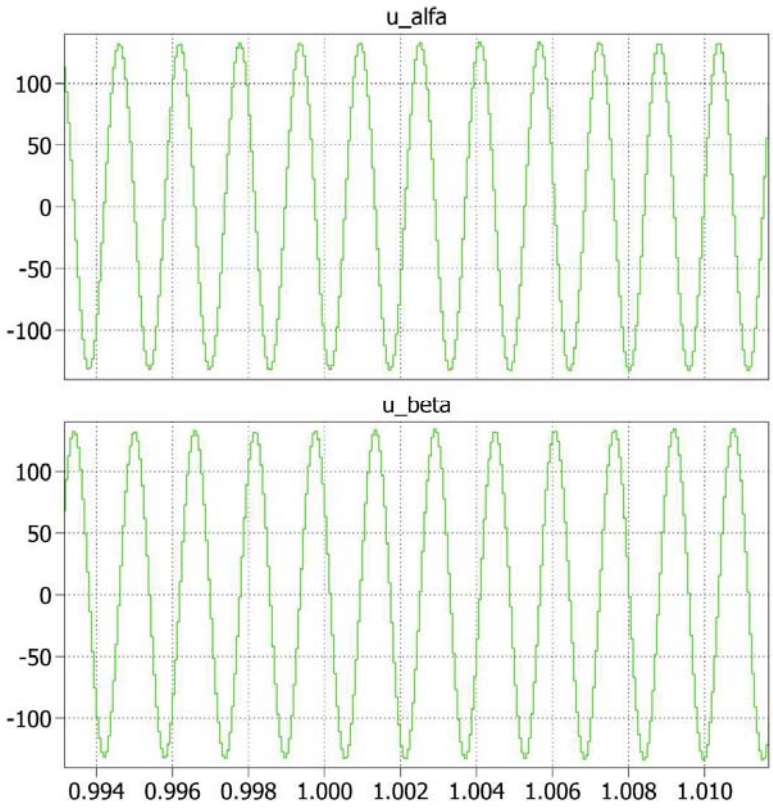


Figure 4.5: linear zone

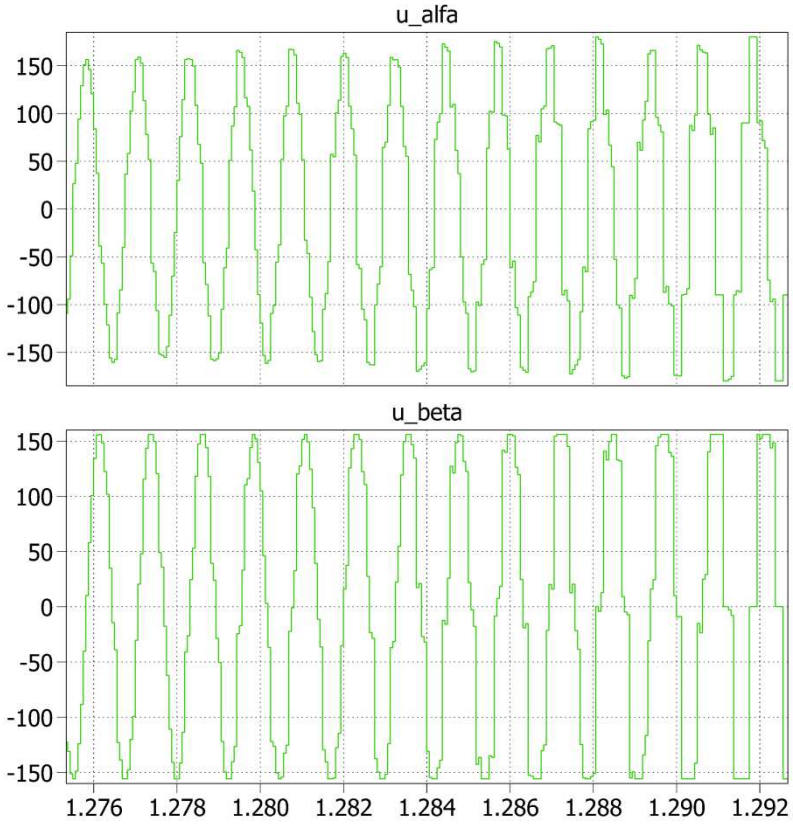


Figure 4.6: overmodulation

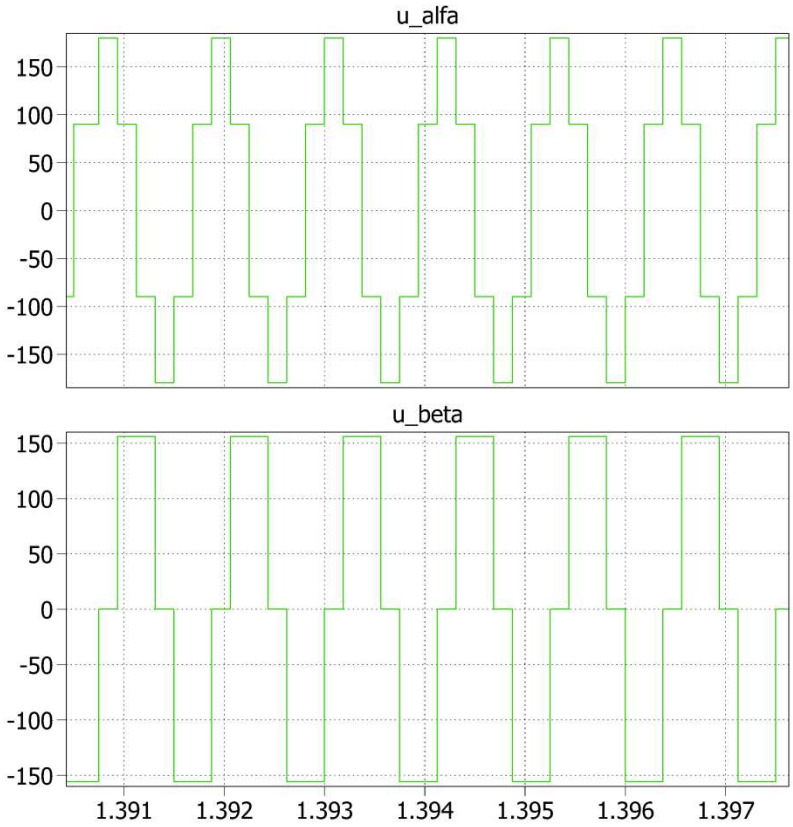


Figure 4.7: Square wave

The alfa-beta voltages in a X-Y plot represent very well what our algorithm does like we can see in figure 4.8.

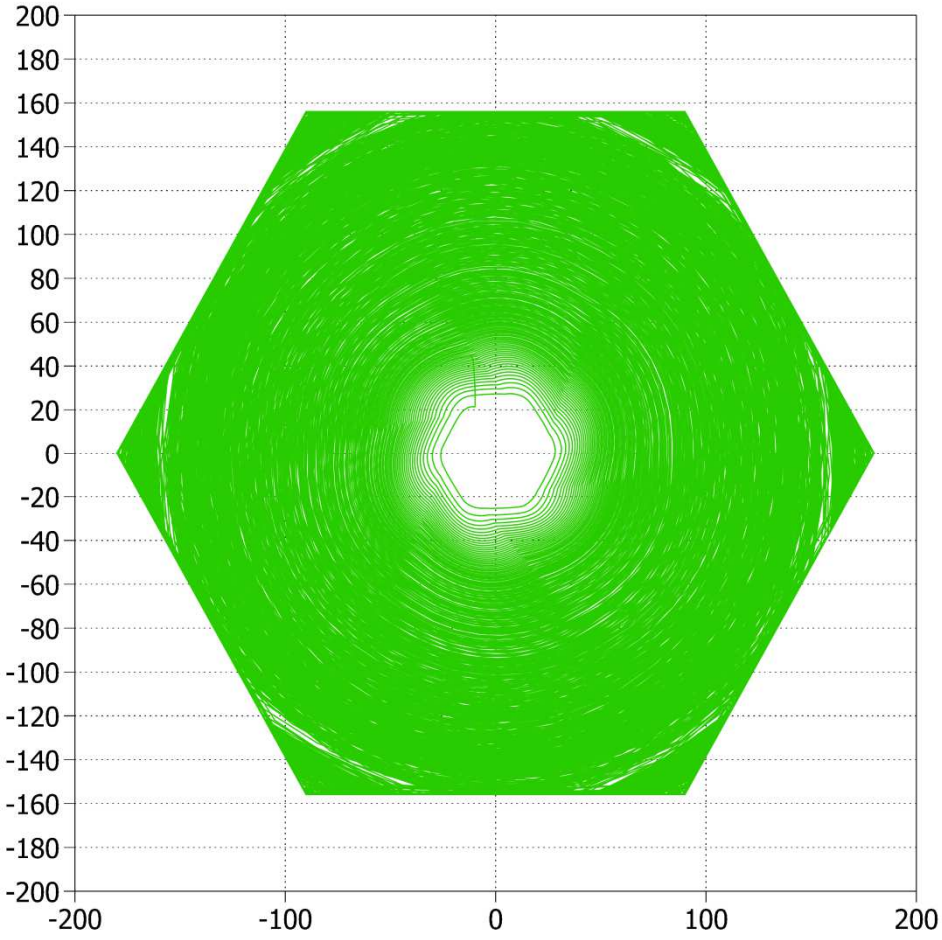


Figure 4.8

### *4.3 Harmonic voltage content analytical description*

Controlling the inverter in overmodulation and six step, we own more voltage from the DC-link, but we have a worse harmonic spectrum and moreover in six step we lose the ability to control the amplitude of the inverter voltage reference. So can be interesting doing an analytical description of the voltage harmonic content. The space vector can be expressed in Fourier series like follows:

$$\boldsymbol{u}(\theta) = \sum_{n\to-\infty}^{+\infty} \boldsymbol{U}_n e^{jn\xi_1} = \sum_{n\to-\infty}^{+\infty} \boldsymbol{U}_n e^{jn\omega_1 t} \qquad (4.23)$$

Where

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{2\pi} \boldsymbol{u}(\theta)\, e^{-jn\xi_1}\, d\xi_1 \qquad (4.24)$$

$$\xi_1 = \omega_1 t \qquad (4.25)$$

And $\omega_1$ is the angle speed of the fundamental.

When $|\boldsymbol{u}(\theta)|$ is constant and the phase is redundant like it is in our algorithm we can write:

$$\theta(\xi_1) = \xi_1 + \Delta\theta(\xi_1) \qquad (4.26)$$

Where $\theta(\xi_1)$ and $\Delta\theta(\xi_1)$ are showed in the followings figures in function of $\xi_1$.

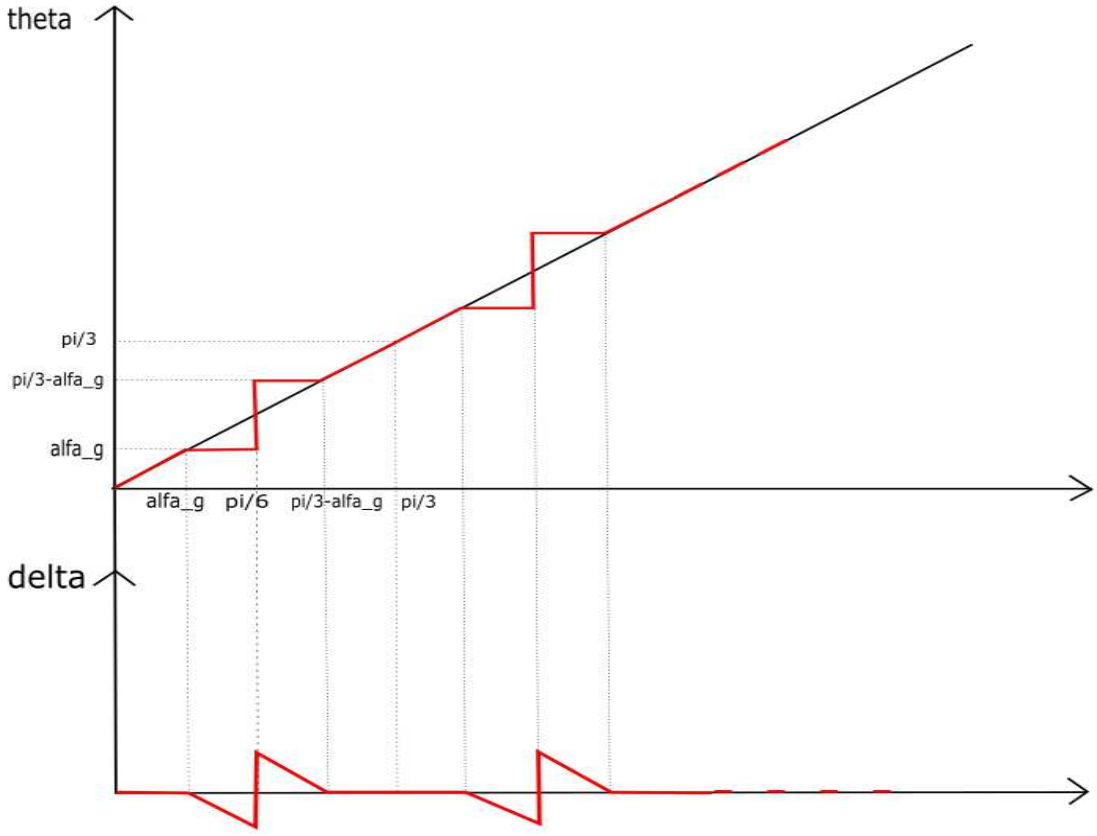


Figure 4.9

Well

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{2\pi} |\boldsymbol{u}(\theta)| e^{j\theta}\, e^{-jn\xi_1}\, d\xi_1 \qquad (4.27)$$

Become

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{2\pi} r\, e^{j\xi_1}\, e^{j\Delta\theta}\, e^{-jn\xi_1}\, d\xi_1 \qquad (4.28)$$

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{2\pi} r\ e^{j\Delta\theta}\, e^{j(1-n)\xi_1}\, d\xi_1 \qquad (4.29)$$

Now using the symmetry of the six sectors of the hexagon we have

$$\boldsymbol{U}_n = \frac{1}{2\pi}\left\{\int_0^{\frac{\pi}{3}} r\ e^{j\Delta\theta}\, e^{j(1-n)\xi_1}\, d\xi_1 + \int_{\frac{\pi}{3}}^{\frac{2\pi}{3}} r\ e^{j\Delta\theta}\, e^{j(1-n)\xi_1}\, d\xi_1 + \cdots + \int_{\frac{5\pi}{3}}^{2\pi} r\ e^{j\Delta\theta}\, e^{j(1-n)\xi_1}\, d\xi_1\right\} \qquad (4.30)$$

$$\boldsymbol{U}_n = \frac{1}{2\pi}\sum_{k=0}^{5}\left\{\int_{\frac{k\pi}{3}}^{\frac{(k+1)\pi}{3}} r\ e^{j\Delta\theta}\, e^{j(1-n)\xi_1}\, d\xi_1\right\} \qquad (4.31)$$

Now we set

$$\xi_2 = \xi_1 - \frac{k\pi}{3} \qquad (4.32)$$

So

$$\xi_1 = \xi_2 + \frac{k\pi}{3} \qquad (4.33)$$

$$d\xi_1 = d\xi_2 \qquad (4.34)$$

And if $\xi_1 = \frac{k\pi}{3}$ then $\xi_2 = 0$ , and if $\xi_1 = \frac{(k+1)\pi}{3}$ then $\xi_2 = \frac{\pi}{3}$, therefore

$$\boldsymbol{U}_n = \frac{1}{2\pi}\sum_{k=0}^{5}\left\{\int_0^{\frac{\pi}{3}} r\ e^{j\Delta\theta}\, e^{j(1-n)(\xi_2+\frac{k\pi}{3})}\, d\xi_2\right\} \qquad (4.35)$$

$$\boldsymbol{U}_n = \frac{1}{2\pi}\sum_{k=0}^{5}\left\{e^{\frac{jk\pi}{3}(1-n)}\int_0^{\frac{\pi}{3}} r\ e^{j\Delta\theta}\, e^{j(1-n)\xi_2}\, d\xi_2\right\} \qquad (4.36)$$

But inside the integral there isn't the *k*

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{\frac{\pi}{3}} r\ e^{j(\Delta\theta+\xi_2)}\, e^{-jn\xi_2}\, d\xi_2 \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)} \qquad (4.37)$$

Now we can call

$$\xi_2 = \xi_1 \qquad (4.38)$$

Because the integral doesn't change and finally we have obtained

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{\frac{\pi}{3}} r\ e^{j(\Delta\theta+\xi_1)}\, e^{-jn\xi_1}\, d\xi_1 \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)} \qquad (4.39)$$

But

$$r\ e^{j(\Delta\theta+\xi_1)} = r\ e^{j\theta} = \boldsymbol{u}(\theta) \qquad (4.40)$$

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{\frac{\pi}{3}} \boldsymbol{u}(\theta)\, e^{-jn\xi_1}\, d\xi_1 \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)} \qquad (4.41)$$

Now that we have obtained this expression, we remember that, using our algorithm, in the first sector $\theta$ is equal to

$$\theta = \begin{cases} \theta_1 & for\ \ 0 \le \theta_1 \le \alpha_g \ \ and\ \ \frac{\pi}{3} - \alpha_g \le \theta_1 \le \frac{\pi}{3} \\ \alpha_g & for\ \ \alpha_g \le \theta_1 \le \frac{\pi}{6} \\ \frac{\pi}{3} - \alpha_g & for\ \frac{\pi}{6} \le \theta_1 \le \frac{\pi}{3} - \alpha_g \end{cases} \qquad (4.42)$$

And substituting it in the following

$$\boldsymbol{U}_n = \frac{1}{2\pi}\int_0^{\frac{\pi}{3}} r\ e^{j\theta}\, e^{-jn\xi_1}\, d\xi_1 \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)} \qquad (4.43)$$

We obtain

$$\boldsymbol{U}_n = \frac{1}{2\pi}\left[\int_0^{\alpha_g} r\ e^{j\theta_1}\, e^{-jn\xi_1}\, d\xi_1 + \int_{\alpha_g}^{\frac{\pi}{6}} r\ e^{j\alpha_g}\ e^{-jn\xi_1}\, d\xi_1 + \int_{\frac{\pi}{6}}^{\frac{\pi}{3}-\alpha_g} r\ e^{j\frac{\pi}{3}-\alpha_g}\ e^{-jn\xi_1}\, d\xi_1 + \int_{\frac{\pi}{3}-\alpha_g}^{\frac{\pi}{3}} r\ e^{j\theta_1}\, e^{-jn\xi_1}\, d\xi_1\right] \times \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)} \qquad (4.44)$$

Now we develop the integrals in the square brackets individually observing that $\theta_1 = \xi_1$, then

$$\int_0^{\alpha_g} r\ e^{j\theta_1}\, e^{-jn\xi_1}\, d\xi_1 = \int_0^{\alpha_g} r\ \ e^{j(1-n)\xi_1}\, d\xi_1 = \frac{r}{j(1-n)}\left(e^{j(1-n)\alpha_g} - 1\right) \qquad (4.45)$$

$$\int_{\alpha_g}^{\frac{\pi}{6}} r\ e^{j\alpha_g}\ e^{-jn\xi_1}\, d\xi_1 = -\frac{re^{j\alpha_g}}{jn}\left(e^{-\frac{jn\pi}{6}} - e^{-jn\alpha_g}\right)$$
$$= -\frac{r}{jn}\left[e^{j\left(\alpha_g - n\frac{\pi}{6}\right)} - e^{j(1-n)\alpha_g}\right] \qquad (4.46)$$

$$\int_{\frac{\pi}{6}}^{\frac{\pi}{3}-\alpha_g} r\ e^{j\frac{\pi}{3}-\alpha_g}\ e^{-jn\xi_1}\, d\xi_1 = -\frac{re^{j\left(\frac{\pi}{3}-\alpha_g\right)}}{jn}\left(e^{-jn\left(\frac{\pi}{3}-\alpha_g\right)} - e^{-jn\frac{\pi}{6}}\right)$$
$$= -\frac{r}{jn}\left[e^{j(1-n)\left(\frac{\pi}{3}-\alpha_g\right)} - e^{j\left(\frac{\pi}{3}-\alpha_g - \frac{n\pi}{6}\right)}\right] \qquad (4.47)$$

$$\int_{\frac{\pi}{3}-\alpha_g}^{\frac{\pi}{3}} r\ e^{j\theta_1}\, e^{-jn\xi_1}\, d\xi_1 = \int_{\frac{\pi}{3}-\alpha_g}^{\frac{\pi}{3}} r\ e^{j(1-n)\xi_1}\, d\xi_1$$
$$= \frac{r}{j(1-n)}\left(e^{j(1-n)\frac{\pi}{3}} - e^{j(1-n)\left(\frac{\pi}{3}-\alpha_g\right)}\right) \qquad (4.48)$$

Substituting these integrals in the complete formula we have

$$\boldsymbol{U}_n = \boldsymbol{U}_n\left(r, \alpha_g\right)$$
$$= \frac{1}{2\pi}\left[\frac{r}{j(1-n)}\left(e^{j(1-n)\alpha_g} - 1\right) - \frac{r}{jn}\left[e^{j\left(\alpha_g - n\frac{\pi}{6}\right)} - e^{j(1-n)\alpha_g}\right]\right.$$
$$\left. - \frac{r}{jn}\left[e^{j(1-n)\left(\frac{\pi}{3}-\alpha_g\right)} - e^{j\left(\frac{\pi}{3}-\alpha_g - \frac{n\pi}{6}\right)}\right] + \frac{r}{j(1-n)}\left(e^{j(1-n)\frac{\pi}{3}} - e^{j(1-n)\left(\frac{\pi}{3}-\alpha_g\right)}\right)\right]$$
$$\times \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)} \qquad (4.49)$$

$$\boldsymbol{U}_n = \boldsymbol{U}_n\left(r, \alpha_g\right)$$
$$= \frac{r}{2\pi j} \times \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-n)}$$
$$\times \left\{\frac{1}{1-n}\left[\left(e^{j(1-n)\alpha_g} - 1\right) + \left(e^{j(1-n)\frac{\pi}{3}} - e^{j(1-n)\left(\frac{\pi}{3}-\alpha_g\right)}\right)\right]\right.$$
$$\left. - \frac{1}{n}\left[e^{j\left(\alpha_g - n\frac{\pi}{6}\right)} - e^{j(1-n)\alpha_g} + e^{j(1-n)\left(\frac{\pi}{3}-\alpha_g\right)} - e^{j\left(\frac{\pi}{3}-\alpha_g - \frac{n\pi}{6}\right)}\right]\right\} \qquad (4.50)$$

This is the final formula that expresses the space vector of the nth harmonic in function of *r* (that is the amplitude of the complete space vector) and of the angle $\alpha_g$ that is equal to

$$\alpha_g = \theta - \frac{\pi}{3} = \arcsin\left(\frac{1}{\sqrt{3}}\frac{U_{DC-link}}{r}\right) - \frac{\pi}{3} \qquad (4.51)$$

Now it's interesting calculating the space vector of the fundamental to understand how is the behaviour during the overmodulation zone.

If $n = 1$ we have:

$$\boldsymbol{U}_1 = \frac{1}{2\pi}\int_0^{\frac{\pi}{3}} \boldsymbol{u}(\theta)\, e^{-j\xi_1}\, d\xi_1 \sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-1)} \qquad (4.52)$$

But

$$\sum_{k=0}^{5} e^{\frac{jk\pi}{3}(1-1)} = \sum_{k=0}^{5} e^{\frac{jk\pi}{3}0} = 6 \qquad (4.53)$$

So

$$\begin{aligned}
\boldsymbol{U}_1 &= \frac{6}{2\pi}\int_0^{\frac{\pi}{3}} \boldsymbol{u}(\theta)\, e^{-j\xi_1}\, d\xi_1 \\
&= \frac{6}{2\pi}\left[\int_0^{\alpha_g} r\ e^{j(\xi_1-\xi_1)}\, d\xi_1 + \int_{\alpha_g}^{\frac{\pi}{6}} r\ e^{j(\alpha_g-\xi_1)}\, d\xi_1\right. \\
&\left. + \int_{\frac{\pi}{6}}^{\frac{\pi}{3}-\alpha_g} r\ e^{j\left(\frac{\pi}{3}-\alpha_g\right)} e^{-jn\xi_1}\, d\xi_1 + \int_{\frac{\pi}{3}-\alpha_g}^{\frac{\pi}{3}} r\ e^{j(\xi_1-\xi_1)}\, d\xi_1\right] \\
&= \frac{6}{2\pi}\left[r\alpha_g + r\, e^{j\alpha_g} j\left(e^{-\frac{j\pi}{6}} - e^{-j\alpha_g}\right) + r\ e^{j\left(\frac{\pi}{3}-\alpha_g\right)} j\left(e^{-j\left(\frac{\pi}{3}-\alpha_g\right)} - e^{-\frac{j\pi}{6}}\right)\right. \\
&\left. + r\alpha_g\right] \qquad (4.54) \\
&= \frac{3r}{\pi}\left[2\alpha_g + j\ e^{j(\alpha_g-\frac{\pi}{6})} - je^{j(\alpha_g-\alpha_g)} + je^{j(\frac{\pi}{3}-\alpha_g-\frac{\pi}{3}+\alpha_g)}\right. \\
&\left. - je^{j(\frac{\pi}{3}-\alpha_g-\frac{\pi}{6})}\right] \qquad (4.55)
\end{aligned}$$

$$= \frac{3r}{\pi}\left[2\alpha_g + 2j\frac{e^{-j(\frac{\pi}{6}-\alpha_g)} - e^{j(\frac{\pi}{6}-\alpha_g)}}{2}\right] \qquad (4.56)$$

$$= \frac{6r}{\pi}\left[\alpha_g + \frac{e^{j(\frac{\pi}{6}-\alpha_g)} - e^{j-(\frac{\pi}{6}-\alpha_g)}}{2j}\right] \qquad (4.57)$$

And using Eulero we obtain:

$$\boldsymbol{U}_1 = \frac{6r}{\pi}[\alpha_g + \sin(\frac{\pi}{6} - \alpha_g)] \qquad (4.58)$$

We can observe that is real so

$$|\boldsymbol{U}_1| = U_1 \qquad (4.59)$$

And finally we have the following expression for the amplitude of the space vector of the fundamental 4.60.

$$U_1 = \frac{6r}{\pi}[\alpha_g \ + \sin(\frac{\pi}{6} - \alpha_g)] \qquad (4.60)$$

In the paper of Prof. Bolognani, it's observed that since the voltage gain characteristic of the SVM (Space Vector Modulation) with the proposed technique is almost linear, a least-squares linear approximation is obtained and it's the following.

$$U_1(r) = 0.707r + 0.169U_{DC} \qquad (4.61)$$

Now, remembering that

$$\alpha_g = \arcsin\left(\frac{1}{\sqrt{3}}\frac{U_{DC-link}}{r}\right) - \frac{\pi}{3} \qquad (4.62)$$

And substituting it in 4.60 we have:

$$U_1(r) = \frac{6r}{\pi}\left[\arcsin\left(\frac{1}{\sqrt{3}}\frac{U_{DC-link}}{r}\right) - \frac{\pi}{3} + \sin\left(\frac{\pi}{6} - arcsin\left(\frac{1}{\sqrt{3}}\frac{U_{DC-link}}{r}\right) + \frac{\pi}{3}\right)\right] \qquad (4.63)$$

We can normalize the two expressions of $U_1(r)$ with the DC-link voltage and write them in function of the amplitude *r* normalized either for the DC-link voltage $U_{DC-link}$ and we have:

$$\frac{U_1}{U_{DC-link}} = \frac{0.707r}{U_{DC-link}} + 0.169 \qquad (4.64)$$

And

$$\frac{U_1}{U_{DC-link}} = \frac{6r}{\pi U_{DC-link}}\left[\arcsin\left(\frac{1}{\sqrt{3}}\frac{U_{DC-link}}{r}\right) - \frac{\pi}{3} + \sin\left(\frac{\pi}{6} - arcsin\left(\frac{1}{\sqrt{3}}\frac{U_{DC-link}}{r}\right) + \frac{\pi}{3}\right)\right] \qquad (4.65)$$

Now imposing for both curves the same initial value point (corresponding to $\frac{r}{U_{DC-link}} = 0.577$ , i.e. the onset of overmodulation) we have the following graphic 4.10.

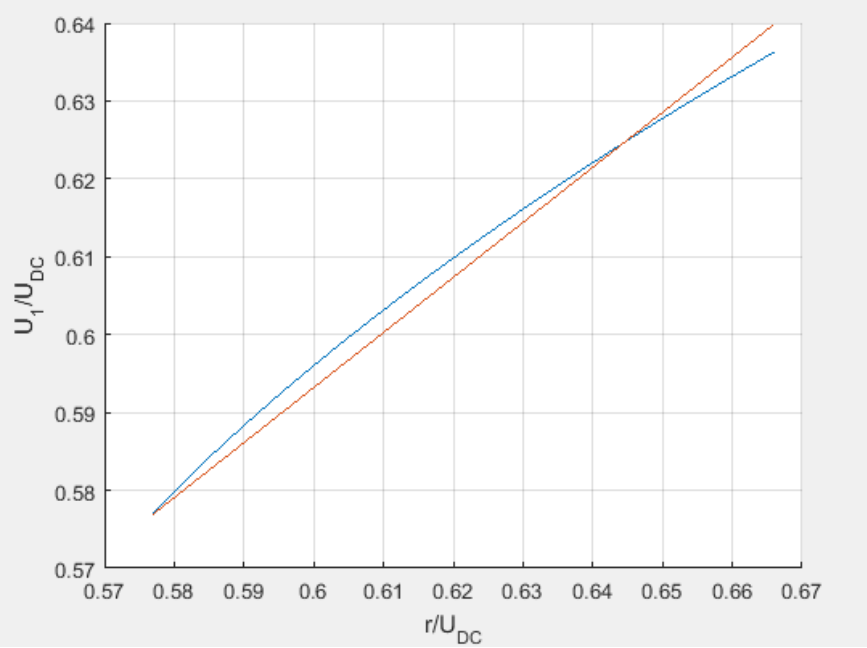


Figure 4.10

The red one is the approximation and the blue one is the ''original'' and the maximum percentage error found is of 0.5% for $r = \frac{2}{3} U_{DC}$.

### *4.4 Square wave and selective harmonic cancellation*

When we are in the condition of square wave 4.12, 4.13 and 4.13 the reference space vector is moving only in the six vertices of the hexagon like we can observe in the figure 4.8. In this section we will explain how the low frequency harmonics caused by the square wave modulation can be neglected.

Let's consider the three-phase inverter in the underlying figure.

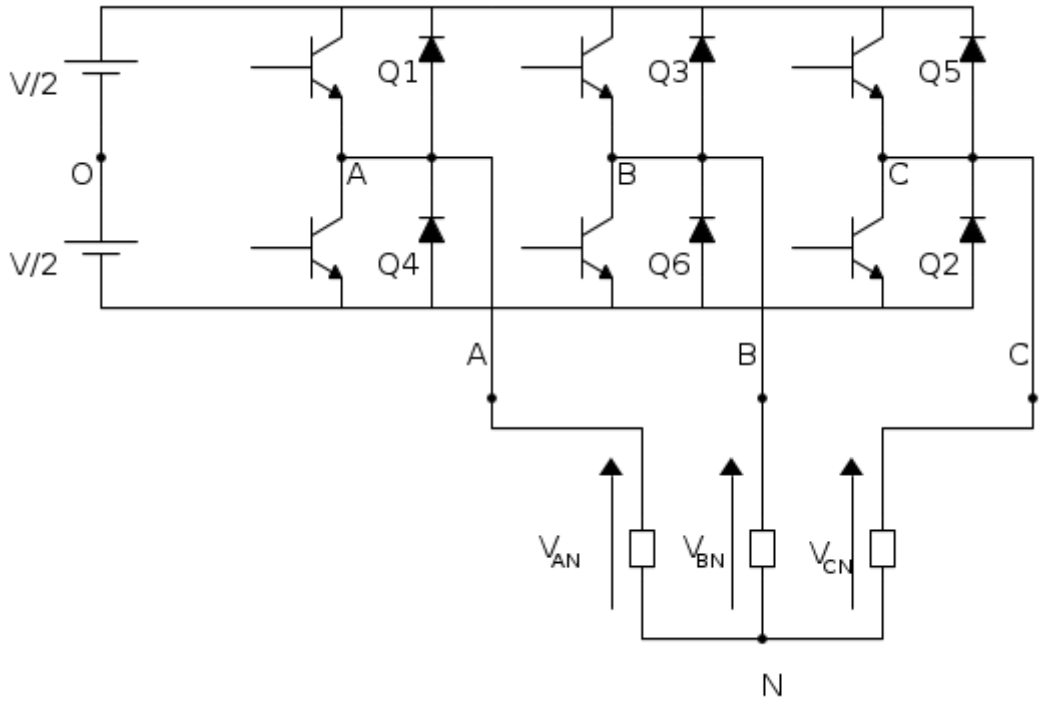


Figure 4.11

When we control the inverter in square wave, we have harmonics in low frequency like the $5^{th}$, the $7^{th}$, $11^{th}$, $13^{th}$ ecc. In the paper titled ''Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverter: Part 1-Harmonic elimination'' of Hasmukh S. Patel and Richard G. Hoft[2], it's proposed a very elegant technique to cancel the harmonics of a square wave in an analytical and selective way.

It is possible to eliminate as many harmonics as the number of pulses per half-cycle of the waveform by constraining the size (in terms of width) and the position of the pulses.

We define M the number of pulses per half-cycle, so the number of commutation per cycle is:

$$N = 2 * 2 * M = 4M \qquad (4.66)$$

The following figure can clear the idea about the pulses in a square wave

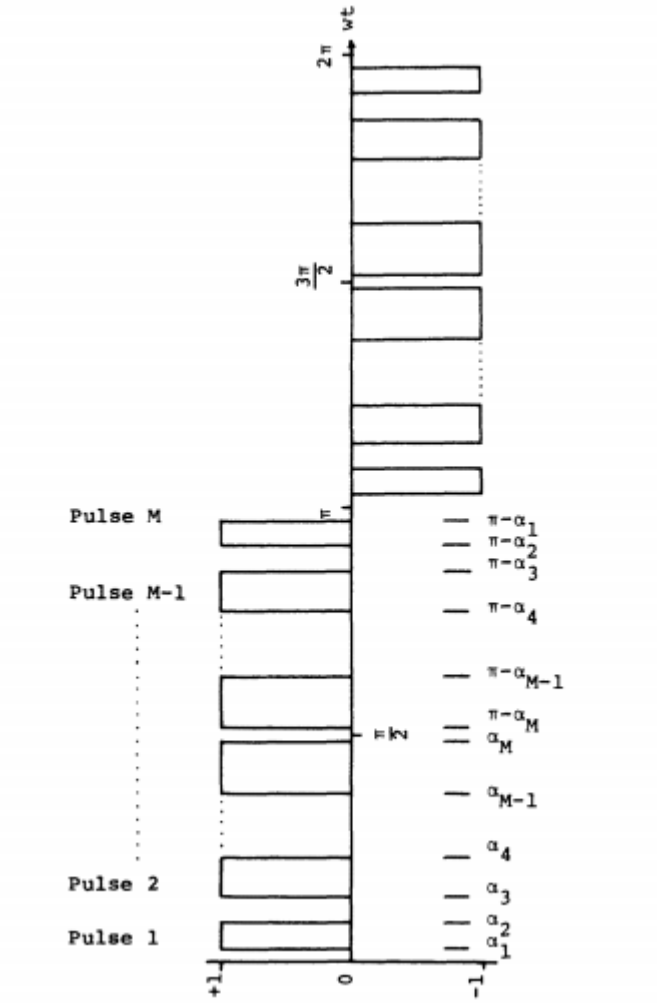


Figure 4.12

The waveform can be represented by a Fuorier Series as follows:

$$f(\omega t) = \sum_{n=1}^{\infty} [a_n \sin(n\omega t) + b_n \cos(n\omega t)] \qquad (4.67)$$

Where

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(\omega t) \sin(n\omega t)\, d(\omega t) \qquad (4.68)$$

and

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(\omega t) \cos(n\omega t)\, d(\omega t) \qquad (4.69)$$

but since our waveform has odd quarter-wave symmetry, the Fourier coefficients are given by:

$$a_n = \frac{4}{\pi}\int_0^{\frac{\pi}{2}} f(\omega t)\sin(n\omega t)\, d(\omega t) \qquad (4.70)$$

for odd *n*

and

$$a_n = 0 \qquad (4.71)$$

for even *n*, while

$$b_n = 0 \qquad (4.72)$$

for all *n*, then we have:

$$f(\omega t) = \sum_{n=1}^{\infty} a_n \sin(n\omega t) \qquad (4.73)$$

Now we observe that from the quarter-wave symmetry property we have the following relation for the chopping angles:

$$\alpha_k = \pi - \alpha_{2M-k+1} \qquad (4.74)$$

With

$$k = 1,2,3 \ldots \ldots M \qquad (4.75)$$

Now we develop the formula of $a_n$ coefficients in function of the commutation angles $\alpha_k$:

$$a_n = \frac{4}{\pi}\int_0^{\frac{\pi}{2}} f(\omega t)\sin(n\omega t)\, d(\omega t) \qquad (4.76)$$

But $f(\omega t)$ can be only 0 or 1 so we have

$$a_n = \frac{4}{\pi}\left[\int_{\alpha_1}^{\alpha_2} \sin(n\omega t)\, d(\omega t) + \int_{\alpha_3}^{\alpha_4} \sin(n\omega t)\, d(\omega t) + \ldots\ldots + \int_{\alpha_{M-1}}^{\alpha_M} \sin(n\omega t)\, d(\omega t) + \right]$$

$$= \frac{4}{\pi}\left[-\frac{1}{n}\cos(n\omega t)\,|_{\alpha_1}^{\alpha_2} - \frac{1}{n}\cos(n\omega t)\,|_{\alpha_3}^{\alpha_4} - \cdots - \frac{1}{n}\cos(n\omega t)\,|_{\alpha_{M-1}}^{\alpha_M}\right]$$

$$= \frac{4}{n\pi}[\cos(n\alpha_1) - \cos(n\alpha_2) + \cos(n\alpha_3) - \cos(n\alpha_4) + \cdots + \cos(n\alpha_{M-1}) - \cos(n\alpha_M)] \qquad (4.77)$$

So

$$a_n = \frac{4}{n\pi}\sum_{k=1}^{M}(-1)^{k+1}\cos(n\alpha_k) \qquad (4.78)$$

Where

$$0 < \alpha_1 < \alpha_2 < \alpha_3 < \cdots < \alpha_{M-1} < \alpha_M < \frac{\pi}{2} \qquad (4.79)$$

With the final formula, we have obtained the harmonics amplitude in function of the commutation angles in an analytical way, that is a very elegant way to resolve problems.

Now that we have written the amplitudes of all the harmonics in function of the angles of switching, we want to see how the harmonics can be eliminated. The approach proposed in the paper is to set to zero the $a_n$ so we have:

$$f_i(\alpha) = \frac{4}{n_i\pi}\sum_{k=1}^{M}(-1)^{k+1}\cos(n_i\alpha_k) = 0 \qquad (4.80)$$

for $i = 1,2,3 \ldots \ldots M$

that is a system of M non-linear equations and the unique solution is the vector $(\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{M-1}, \alpha_M)$ therefore M harmonics are eliminated. To resolve that system, the paper suggest to use a numerical method, but we have thought at another approach to find the vector $(\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{M-1}, \alpha_M)$.

Other way

Instead to resolve a non-linear equation system problem, we can resolve an optimisation problem. To resolve the problem in this way we have to find a proper function therefore the vector that gives the minimum is our solution. If we want to eliminate the 5$^{th}$ and the 7$^{th}$ harmonics, our function can be:

$$F(\alpha_1, \alpha_2) = |a_5| + |a_7| \qquad (4.81)$$

In this way if we find the minimum of the function, we obtain the vector $(\alpha_1, \alpha_2)$ therefore F is minimum and so we find $(\alpha_1, \alpha_2)$ therefore

$$a_5 = a_7 = 0 \qquad (4.82)$$

Problem 1

How can we do in order to obtain a solution $(\alpha_1, \alpha_2)$ that fulfils the following restriction?

$$0 < \alpha_1 < \alpha_2 < \frac{\pi}{2} \qquad (4.83)$$

Solution problem1

If we want that

$$0 < \alpha_1 < \alpha_2 < \frac{\pi}{2} \qquad (4.84)$$

We can write a new function G

$$G(\alpha_1, \alpha_2) = F(\alpha_1, \alpha_2) + t_1(\alpha_1, \alpha_2) \qquad (4.85)$$

Where *t1* is the expression that permits to respect the restriction

$$t_1(\alpha_1, \alpha_2) = k * sum(abs\left((\alpha_1, \alpha_2) - sort\big((\alpha_1, \alpha_2)\big)\right) \qquad (4.86)$$

*Sort* is a matlab function, for example sort(X) sorts the elements of the vector X in ascending order, and *k* is a scalar coefficient, it is like a weight if necessary.

So until now we have

$$G(\alpha_1, \alpha_2) = |a_5| + |a_7| + k * sum(abs\left((\alpha_1, \alpha_2) - sort\big((\alpha_1, \alpha_2)\big)\right) \qquad (4.87)$$

Problem 2

With only the correction function $t_1(\alpha_1, \alpha_2)$ could be that

$$\alpha_1 = \alpha_2 \qquad (4.88)$$

But we don't want it

Solution problem2

This solution is not general because it resolves the problem only if we have 2 angles, but for now it is enough in fact we want to eliminate the 5$^{th}$ and the 7$^{th}$ harmonics only. So we have to write a new function still:

$$H(\alpha_1, \alpha_2) = G(\alpha_1, \alpha_2) + t_2(\alpha_1, \alpha_2) \qquad (4.89)$$

Where

$$t_2(\alpha_1, \alpha_2) = \frac{1}{1 + |\alpha_1 - \alpha_2|} \qquad (4.90)$$

In fact we are searching the minimum of the function, so our algorithm will find a solution therefore *t2* is minimum, but we can easily see that *t2* is max when

$$\alpha_1 = \alpha_2 \qquad (4.91)$$

Then our method will avoid this condition.

Finally the function that we want minimize is:

$$H(\alpha_1, \alpha_2) = F(\alpha_1, \alpha_2) + t_1(\alpha_1, \alpha_2) + t_2(\alpha_1, \alpha_2) \qquad (4..92)$$

Now we have to choose the optimization algorithm to implement in order to obtain $(\alpha_1, \alpha_2)$ therefore the function *H* is minima. During the course of Professor Alotto, we have studied a particular optimization method the DE (Differential Evolution) that is explained in the paper '' Using Differential Evolution for Combinatorial Optimization: A General Approach'' of Ricardo S. Prado Rodrigo C. P. Silva and Frederico G. Guimaraes, Oriane M. Neto[3].

Like other evolutionary algorithms, the original DE algorithm works with a population of candidate solutions randomly generated within the domain region of the problem, usually described as:

$$\mathrm{X} = \left\{x \in R^n : x_k^{\min} \leq x_k \leq x_k^{\max}, k = 1, \dots, n\right\} \qquad (4.93)$$

Where $x_k^{\min}$ and $x_k^{\max}$are respectively the low and upper limits of each variable. they adopt in this paper the notation $x_{g,i,j}$ such that g = 1, ... , G represents the generation counter;
i = 1,...,$\mu$ represents the index of the individual in the population; and j = 1, ... , n represents the variable index. A given individual is represented by $x_{g,i}$. New individuals are generated by using the differential mutation. The mutation is based on the difference between two individuals randomly chosen from the current population. This differential vector is multiplied by a constant and added to a third individual, called base vector (base solution), leading to the so-called mutant vector:

$$v_{g,i} = x_{g,r_1} + F\left(x_{g,r_2} - x_{g,r_3}\right) \qquad (4.94)$$

Where $r_1 \neq r_2 \neq r_3 \in \{1, \dots, \mu\}$ are mutually distinct random indices, and F is a differential weight, a scale factor applied to the differential vector. For each $x_{g,i}$ in the population a corresponding mutant solution $v_{g,i}$ is generated. A trial vector $u_{g,i}$ is produced through recombination of $x_{g,i}$ and $v_{g,i}$. In the basic DE algorithm, the discrete recombination with probability $P_x$ is used. In this way, F and $P_x$ represent control parameters of the algorithm. Finally, the trial vector $u_{g,i}$ competes with the current solution $x_{g,i}$ based on their objective function evaluations. If the trial solution is better or equal than the current solution, it replaces the current solution, otherwise the current solution survives while the trial one is eliminated, as described below:

$$x_{g+1,i} = \begin{cases} u_{g,i} & if\ \ f(u_{g,i}) \leq f(x_{g,i}) \\ x_{g,i} & otherwise \end{cases} \qquad (4.95)$$

In the appendix of this chapter, can be find the code to implement this algorithm of optimization that is very short and easy to understand.

Now let's see the elimination of the 5th and the 7th harmonics that are the most annoying in terms of disturb because they are at low frequency. If we want to eliminate two harmonics, the number of pulses per half-cycle M is:

$$M = 2 \qquad (4.96)$$

The formula of the amplitude of the harmonics in function of the chopping angles we saw to be:

$$a_n = \frac{4}{n\pi}\sum_{k=1}^{M}(-1)^{k+1}\cos(n\alpha_k) \qquad (4.97)$$

So

$$a_5 = \frac{4}{5\pi}\sum_{k=1}^{2}(-1)^{k+1}\cos(5\alpha_k) \qquad (4.98)$$

$$a_7 = \frac{4}{7\pi}\sum_{k=1}^{2}(-1)^{k+1}\cos(7\alpha_k) \qquad (4.99)$$

In this way the function F is:

$$F(\alpha_1, \alpha_2) = |a_5| + |a_7| \qquad (4.100)$$

$$F(\alpha_1, \alpha_2) = \frac{4}{5\pi}|\cos(5\alpha_1) - \cos(5\alpha_2)| + \frac{4}{7\pi}|\cos(7\alpha_1) - \cos(7\alpha_2)| \qquad (4.101)$$

Let's define t_1:

$$t_1(\alpha_1, \alpha_2) = 1 * sum(abs\left((\alpha_1, \alpha_2) - sort\left((\alpha_1, \alpha_2)\right)\right) \qquad (4.102)$$

and t_2:

$$t_2(\alpha_1, \alpha_2) = \frac{1}{1 + |\alpha_1 - \alpha_2|} \qquad (4.103)$$

Well the final function H that we want to minimize is:

$$H(\alpha_1, \alpha_2) = F(\alpha_1, \alpha_2) + t_1(\alpha_1, \alpha_2) + t_2(\alpha_1, \alpha_2) \qquad (4.104)$$

$$H(\alpha_1, \alpha_2) = \frac{4}{5\pi}|\cos(5\alpha_1) - \cos(5\alpha_2)| + \frac{4}{7\pi}|\cos(7\alpha_1) - \cos(7\alpha_2)| + sum(abs\left((\alpha_1, \alpha_2) - sort\left((\alpha_1, \alpha_2)\right)\right) + \frac{1}{1 + |\alpha_1 - \alpha_2|} \qquad (4.105)$$

Now if we apply the DE method to find the minimum of $H(\alpha_1, \alpha_2)$, we find the solution $\bar{x}(\alpha_1, \alpha_2)$

$$\bar{x}(\alpha_1, \alpha_2) = [0.2693\ ; 1.5260] \qquad (4.106)$$

That in degrees is:

$$\bar{x}(\alpha_1^\circ, \alpha_2^\circ) = [15.4297^\circ ; 87.4333^\circ] \qquad (4.107)$$

The solution of the paper ''Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverter: Part 1-Harmonic elimination'' of Hasmukh S. Patel and Richard G. Hoft, was:

$$\overline{x_{paper}}(\alpha_1^\circ, \alpha_2^\circ) = [15.4226^\circ ; 87.3949^\circ] \qquad (4.108)$$

But instead to resolve a system of non- linear equation, we have resolved an optimization problem and the relative final error is of 0.04%.

In the following figures there are the results of a simulation in square wave.

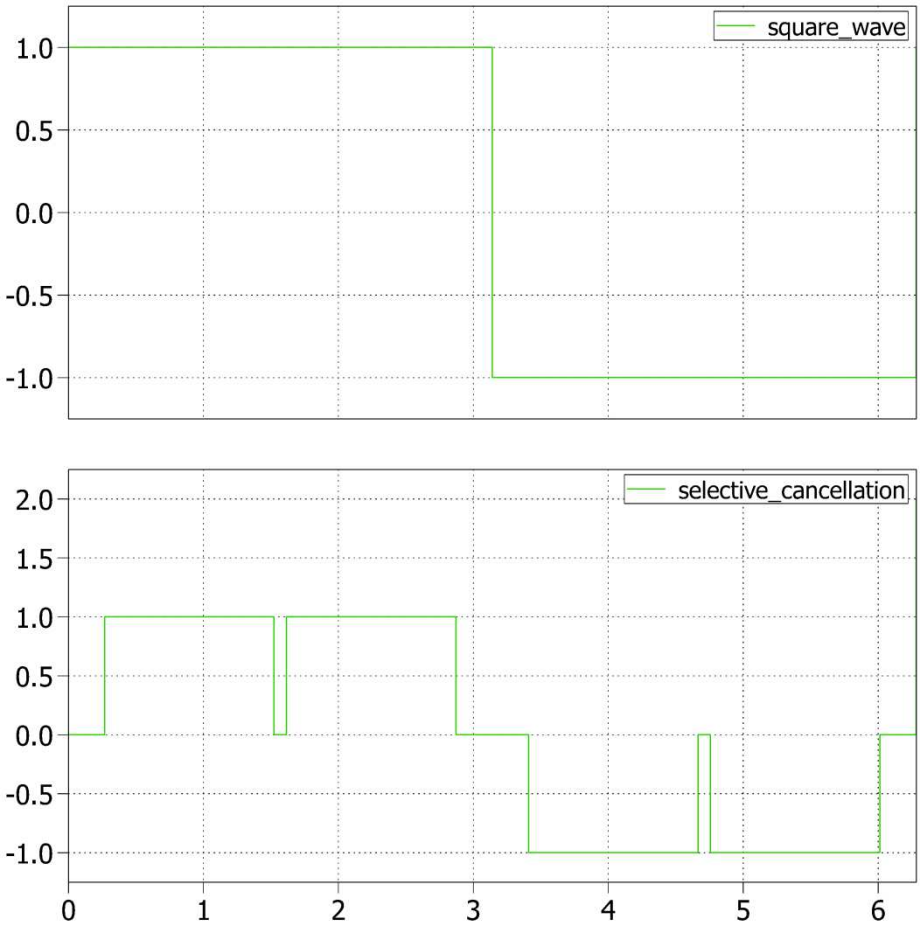


Figure 4.13

The *fft* of the two waves to see if really the cancellation of the 5$^{th}$ and the 7$^{th}$ is happened is showed is figure 4.14:

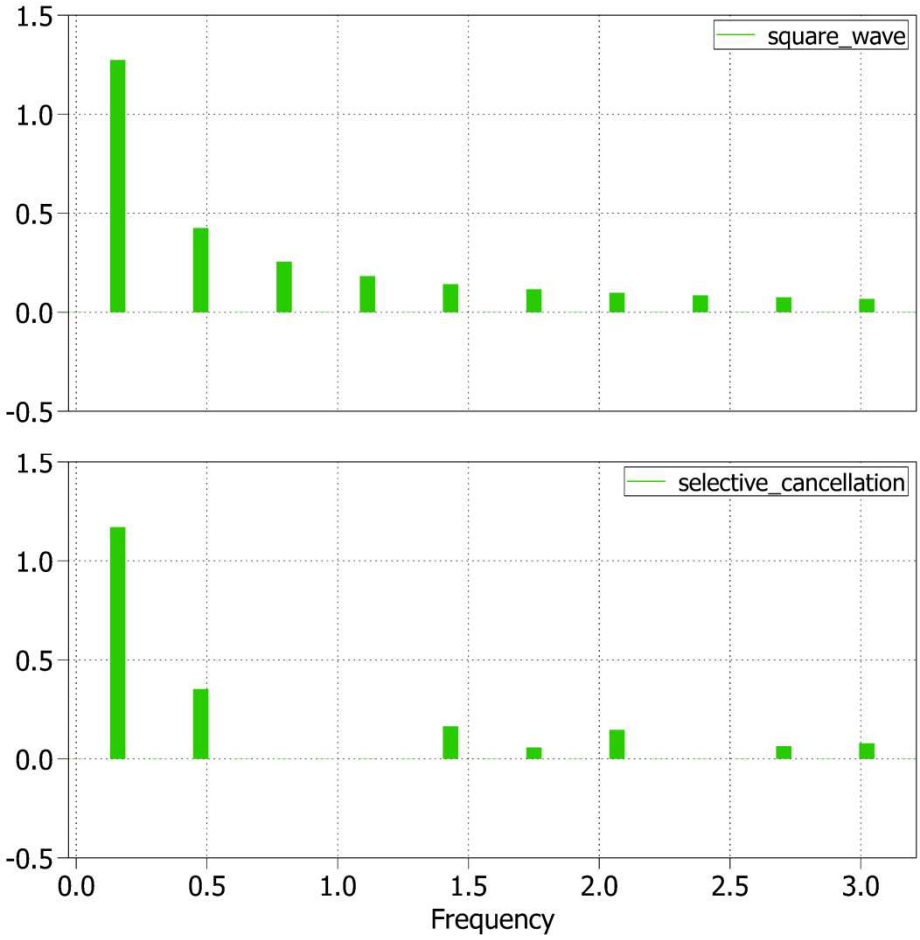


Figure 4.14

Finally let's see how applying the cuts on the square wave in the simulation:

When the speed is constant and we are using a square wave control, the signals that arrive at the inverter legs are square waves with a frequency equals to the frequency of the motor. To make the chops in the square wave, we use a programmed lookup table with the angles calculated before, that makes a wave that summed with the square wave, produces the final wave form that we want. Below there is an example only for the phase A where is showed the scheme to make the chopped wave.

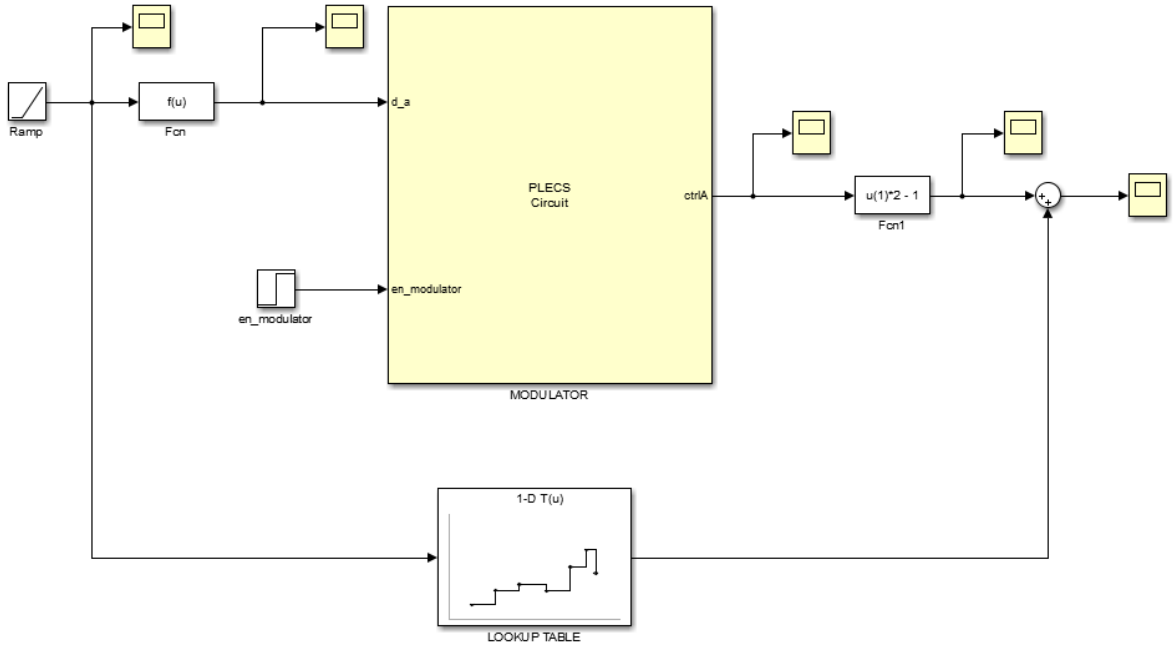


Figure 4.15

### *4.5 Appendix*

Overmodulation code

```
function [u_alfa_new, u_beta_new] = overmodulation(u_alfa, u_beta,
U_lim_LL)
 %%%% questo blocco fa il passaggio continuo da PWM zona lineare a
overmodulation e onda quadra

% calcoliiamo l'angolo teta
u_alfa_new = 0;
u_beta_new = 0;
teta = 0;
 if u_alfa > 0  &&  u_beta >= 0
    teta = atan(u_beta/u_alfa);
end
if u_alfa > 0  &&  u_beta < 0
    teta = atan(u_beta/u_alfa) + 2*pi;
    end
if u_alfa < 0
    teta = atan(u_beta/u_alfa) + pi;
end
if u_alfa == 0  &&  u_beta > 0
    teta = pi/2;
end
if u_alfa == 0  &&  u_beta < 0
    teta = 1.5*pi;
end
if u_alfa == 0 && u_beta == 0
    teta = 0;
end
teta;
 %%%% procediamo con l'algoritmo
 % calcoliamo il modulo di u_alfabeta
if sqrt(u_alfa^2 + u_beta^2) < 2/3*U_lim_LL
    U_alfabeta = sqrt(u_alfa^2 + u_beta^2);
else
        U_alfabeta = 2/3*U_lim_LL;
end
U_alfabeta;
apotema = U_lim_LL/sqrt(3);  % e' l'apotema dell'esagono di lato 1.5U_DC-
link
if U_alfabeta < apotema
    u_alfa_new = u_alfa;
    u_beta_new = u_beta;
else
    % calcoliamo l'angolo alfa_g come l'intersezione della circonferenza di
raggio U_alfabeta e la retta u_beta=apotema dell'esagono
    teta_g = asin(apotema/U_alfabeta);
    alfa_g = teta_g - pi/3;  % vedi i calcoli su foglio
    %% dividiamo la cosa per settori e partiamo dal settore che va da zero
a pigreco terzi
    if teta >= 0  &&  teta <= pi/3
       if teta >= 0  &&  teta <= alfa_g
           u_alfa_new = u_alfa;
           u_beta_new = u_beta;
       end
       if teta > alfa_g  &&  teta <= pi/6
```

```
        u_alfa_new = U_alfabeta*cos(alfa_g);
        u_beta_new = U_alfabeta*sin(alfa_g);
    end
    if teta > pi/6  &&  teta <= (pi/3 - alfa_g)
        u_alfa_new = U_alfabeta*cos(pi/3-alfa_g);
        u_beta_new = U_alfabeta*sin(pi/3-alfa_g);
    end
    if teta > (pi/3 - alfa_g)  &&  teta <= pi/3
        u_alfa_new = u_alfa;
        u_beta_new = u_beta;
    end
end
%% settore che va da pi/3 a 2pi/3
 if teta > pi/3   &&  teta <= 2*pi/3
    if teta > pi/3  &&  teta <= (pi/3 + alfa_g)
        u_alfa_new = u_alfa;
        u_beta_new = u_beta;
    end
    if teta > (pi/3 + alfa_g)  &&  teta <= pi/2
        u_alfa_new = U_alfabeta*cos(pi/3+alfa_g);
        u_beta_new = U_alfabeta*sin(pi/3+alfa_g);
    end
    if teta > pi/2  &&  teta <= (2*pi/3 - alfa_g)
        u_alfa_new = U_alfabeta*cos(2*pi/3-alfa_g);
        u_beta_new = U_alfabeta*sin(2*pi/3-alfa_g);
    end
    if teta > (2*pi/3 - alfa_g)  &&  teta <= 2*pi/3
        u_alfa_new = u_alfa;
        u_beta_new = u_beta;
    end
end
 %% settore che va da 2pi/3 a pi
 if teta > 2*pi/3   &&  teta <= pi
    if teta > 2*pi/3  &&  teta <= (2*pi/3 + alfa_g)
        u_alfa_new = u_alfa;
        u_beta_new = u_beta;
    end
    if teta > (2*pi/3 + alfa_g)  &&  teta <= (pi - pi/6)
        u_alfa_new = U_alfabeta*cos(2*pi/3+alfa_g);
        u_beta_new = U_alfabeta*sin(2*pi/3+alfa_g);
    end
    if teta > (pi - pi/6)  &&  teta <= (pi - alfa_g)
        u_alfa_new = U_alfabeta*cos(pi-alfa_g);
        u_beta_new = U_alfabeta*sin(pi-alfa_g);
    end
    if teta > (pi - alfa_g)  &&  teta <= pi
        u_alfa_new = u_alfa;
        u_beta_new = u_beta;
    end
end
 %% settore che va da pi a pi+pi/3
 if teta > pi   &&  teta <= (pi + pi/3)
    if teta > pi  &&  teta <= (pi + alfa_g)
        u_alfa_new = u_alfa;
        u_beta_new = u_beta;
    end
    if teta > (pi + alfa_g)  &&  teta <= (pi + pi/6)
        u_alfa_new = U_alfabeta*cos(pi+alfa_g);
        u_beta_new = U_alfabeta*sin(pi+alfa_g);
    end
    if teta > (pi + pi/6)  &&  teta <= (pi + pi/3 - alfa_g)
```

```
            u_alfa_new = U_alfabeta*cos(pi + pi/3 - alfa_g);
            u_beta_new = U_alfabeta*sin(pi + pi/3 - alfa_g);
        end
        if teta > (pi + pi/3 - alfa_g)  &&  teta <= (pi + pi/3)
            u_alfa_new = u_alfa;
            u_beta_new = u_beta;
        end
    end
    %% settore che va da pi+pi/3 a 5/3pi
     if teta > (pi + pi/3)   &&  teta <= 5*pi/3
        if teta > (pi + pi/3)  &&  teta <= (pi + pi/3 + alfa_g)
            u_alfa_new = u_alfa;
            u_beta_new = u_beta;
        end
        if teta > (pi + pi/3 + alfa_g)  &&  teta <= 3*pi/2
            u_alfa_new = U_alfabeta*cos(pi+pi/3+alfa_g);
            u_beta_new = U_alfabeta*sin(pi+pi/3+alfa_g);
        end
        if teta > 3*pi/2  &&  teta <= (5*pi/3 - alfa_g)
            u_alfa_new = U_alfabeta*cos(5*pi/3 - alfa_g);
            u_beta_new = U_alfabeta*sin(5*pi/3 - alfa_g);
        end
        if teta > (5*pi/3 - alfa_g)  &&  teta <= 5*pi/3
            u_alfa_new = u_alfa;
            u_beta_new = u_beta;
        end
    end
    %% settore che va da 5/3pi a 2pi
     if teta > 5*pi/3   &&  teta <= 2*pi
        if teta > 5*pi/3  &&  teta <= (5*pi/3 + alfa_g)
            u_alfa_new = u_alfa;
            u_beta_new = u_beta;
        end
        if teta > (5*pi/3 + alfa_g)  &&  teta <= (2*pi - pi/6)
            u_alfa_new = U_alfabeta*cos(5*pi/3+alfa_g);
            u_beta_new = U_alfabeta*sin(5*pi/3+alfa_g);
        end
        if teta > (2*pi - pi/6)  &&  teta <= (2*pi - alfa_g)
            u_alfa_new = U_alfabeta*cos(2*pi - alfa_g);
            u_beta_new = U_alfabeta*sin(2*pi - alfa_g);
        end
        if teta > (2*pi - alfa_g)  &&  teta < 2*pi
            u_alfa_new = u_alfa;
            u_beta_new = u_beta;
        end
    end
end
```

Differential Evolution method DE

```
function [x_new] = DE(x,n_ind,dim,F,px,l,nodo)

 %% mutazione

for i = 1:n_ind
indice = randperm(n_ind,3);
v(:,i) = x(:,indice(1)) + F*(x(:,indice(2)) - x(:,indice(3)));
end

%% ricombinazione

for i = 1:dim
    for j = 1:n_ind
        s = rand(1);
        if s >= px
            u(i,j) = x(i,j);
        else
            u(i,j) = v(i,j);
        end

        % correzione punti usciti dal dominio

        if u(i,j) < nodo(i)
            u(i,j) = nodo(i)+l(i)/3;
        end
        if u(i,j) > nodo(i)+l(i)
            u(i,j) = nodo(i)+l(i)*2/3;
        end
    end
end

%% SELEZIONE - creazione di x_new

for i = 1:n_ind
    if funzione_francesco(x(:,i)) < funzione_francesco(u(:,i)')
        x_new(:,i) = x(:,i);
    else
        x_new(:,i) = u(:,i);
    end
end

end
```

# Chapter 5:

# Proportional-resonant controllers

### *5.1 Introduction*

In this chapter we will describe the proportional-resonant controllers (PR) used in the model to neglect the harmonics caused by the dead time basically. The theory of PR is well written in the papers ‘’Improved Design and Control of Proportional Resonant Controller for Three-Phase Voltage Source Inverter’’ of Mehdi Ebad and Byeong-Mun Song, Department of Electrical and Computer Engineering, Baylor University, Waco, Texas, USA [4] and ‘’ A Novel Current-Tracking Method for Active Filters Based on a Sinusoidal Internal Model’’ of Shoji Fukuda, Senior Member, IEEE, and Takehito Yoda [5]. Using the PR controllers, the converter reference tracking performance can be enhanced and previously known shortcomings associated with conventional PI controllers can be alleviated.

### *5.2 Internal model principle*

The internal model is the theory principle of the proportional-resonant controllers. The internal model principle implies that, in a feedback system, the output of a control object follows its reference input without a steady-state error if the system satisfies the following two conditions:

1) The closed system is asymptotically stable.

2) The open-loop transfer function of the system includes a mathematical model which can generate the required reference input.

Let us take a sinusoidal reference signal as an example and prove that the steady-state error will be zero, if the internal model principle is satisfied. Consider a simple feedback system, where the transfer functions of the compensator and the control object are $G_c(s)$ and $G_r(s)$, respectively. Let $R(s)$, $E(s)$ and $Y(s)$ denote the reference, error, and output signal, respectively. Then, the open-loop transfer function $G_O(s)$ will be given by

$$G_O(s) = G_c(s)G_r(s) = \frac{N_O(s)}{D_O(s)} \qquad (5.1)$$

For the sinusoidal reference

$$r(t) = Acos(\omega_0 t) \qquad (5.2)$$

Appling the Laplace transform

$$R(s) = \frac{As}{s^2 + \omega_0^2} \qquad (5.3)$$

The error between $R(s)$ and $Y(s)$ can be expressed as follows:

$$E(s) = \frac{R(s)}{1 + G(s)} = \frac{D_O(s)R(s)}{D_O(s) + N_O\,(s)} = \frac{D_O(s)}{(s-\mu_1)(s-\mu_1)\dots(s-\mu_1)}\,\frac{As}{s^2+\omega_0^2} \qquad (5.4)$$

assuming, for simplicity of arguments, that the $\mu_i$ are distinct. From condition 1), the poles of the closed system,$\mu_i$ , must satisfy the following equation:

$$Re(\mu_i) \leq 0 \qquad for\ i = 1,2,\dots,m \qquad (5.5)$$

The error can also be expressed as

$$E(s) = \frac{a_1}{s-\mu_1} + \frac{a_2}{s-\mu_2} + \cdots + \frac{a_m}{s-\mu_m} + \frac{b_1}{s-j\omega_0} + \frac{b_2}{s+j\omega_0} \qquad (5.6)$$

where the coefficients $b_1$ and $b_2$ are given by

$$b_{1,2} = [(s \mp j\omega_0)E(s)]_{s=\pm j\omega_0} = \frac{D_O(\pm j\omega_0)}{D_O(\pm j\omega_0) + N_O(\pm j\omega_0)}\,\frac{A}{2} \qquad (5.7)$$

Since $G_O(s)$ has poles $\pm j\omega_0$ from condition 2), the denominator of $G_O(s)$ can be expressed as

$$D_O(s) = (s^2 + \omega_0^2)D_O'(s) \qquad (5.8)$$

it follows that

$$D_O(\pm j\omega_0) = 0 \qquad (5.9)$$

$$N_O(\pm j\omega_0) \neq 0 \qquad (5.10)$$

$$b_{1,2} = 0 \qquad (5.11)$$

Thus, the last two terms in (5.6) disappear, guaranteeing the reduction of the error to zero as time elapses. In a similar manner*, it can be easily proved that the effects of an AC disturbance at the frequency $\omega_0$ can be eliminated.*

### *5.3 PR*

Now that we have described the internal model principle, we can observe in detail the Proportional Resonant current control (PR). The employment of proportional resonant controller, compared with other conventional methods, gives the following advantages:

a) There is zero steady-state error for sinusoidal waveforms having the same frequency as $\omega_0$ ; this feature can be exploited for harmonic compensation (HC), where the signal frequencies are well defined and practically constant (mains' frequency and its multiples).
b) The PR acts as a resonant filter, tuned on $\omega_0$; in this way, multiple PRs with different resonance frequencies can operate in parallel without interfering with each other.

The ideal PR control, which is based on an internal modern theory, is expressed as:

$$G_s = K_p + \frac{2K_i s}{s^2 + \omega_0^2} \qquad (5.12)$$

Where $\omega_0$ is the fundamental frequency and $K_p$ and $K_i$ represent proportional and resonant gains respectively. For $K_p$, it is tuned in the same way as for a PI controller, and it basically determines the dynamics of the system in terms of bandwidth, phase and gain margin and $K_i$ could be tuned for shifting the magnitude response vertically but this does not give rise to a significant variation in bandwidth. For fundamental frequency component, the PR control for the ac quantity in the stationary frame is equal to the PI control for the dc quantity in the synchronous frame. The Bode plots of this ideal PR control method are shown for example in figure 5.1.

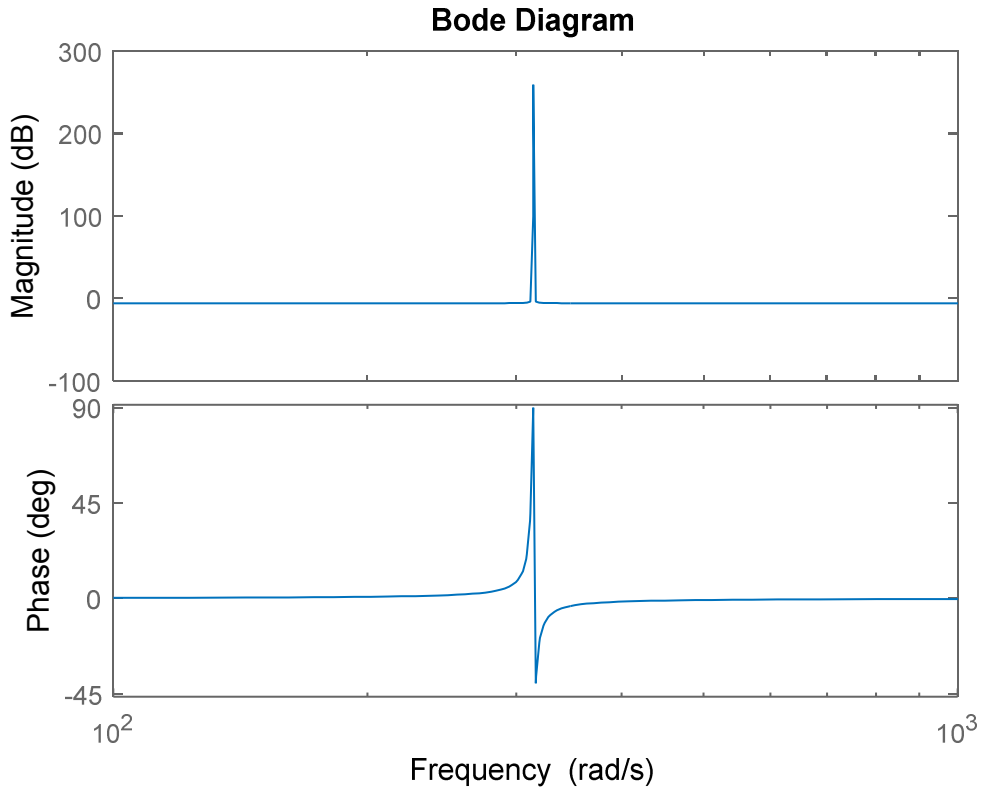


Figure 5.1: ideal PR compensator

As it can be seen in Fig. 5.1, the ideal controller has a theoretically infinite gain at the AC frequency of $\omega_0$ enabling to eliminate the steady-state error to zero and no gain or phase shift at other frequencies. At the selected harmonic frequency, the PR control brings an infinite gain; while at the frequency not belongs to the harmonic range, the proportional resonant control has enough attenuation and the PR control would not affect the frequency not selected. In this case, the PR control realizes the elimination of harmonics and brings little influence to the feature of frequency domain of close loop. As the infinite gain may cause a series of stability problems, non-ideal form can be used instead of 5.12 with transfer function given in 5.13.

$$G_s = K_p + \frac{2K_i\omega_c s}{s^2 + 2\omega_c s + \omega_0^2} \qquad (5.13)$$

Its gain is now finite, but still relatively high for catching small steady-state error and good tracking performance for desired harmonics. The Bode plot of non-ideal PR control is shown in figure 5.2.

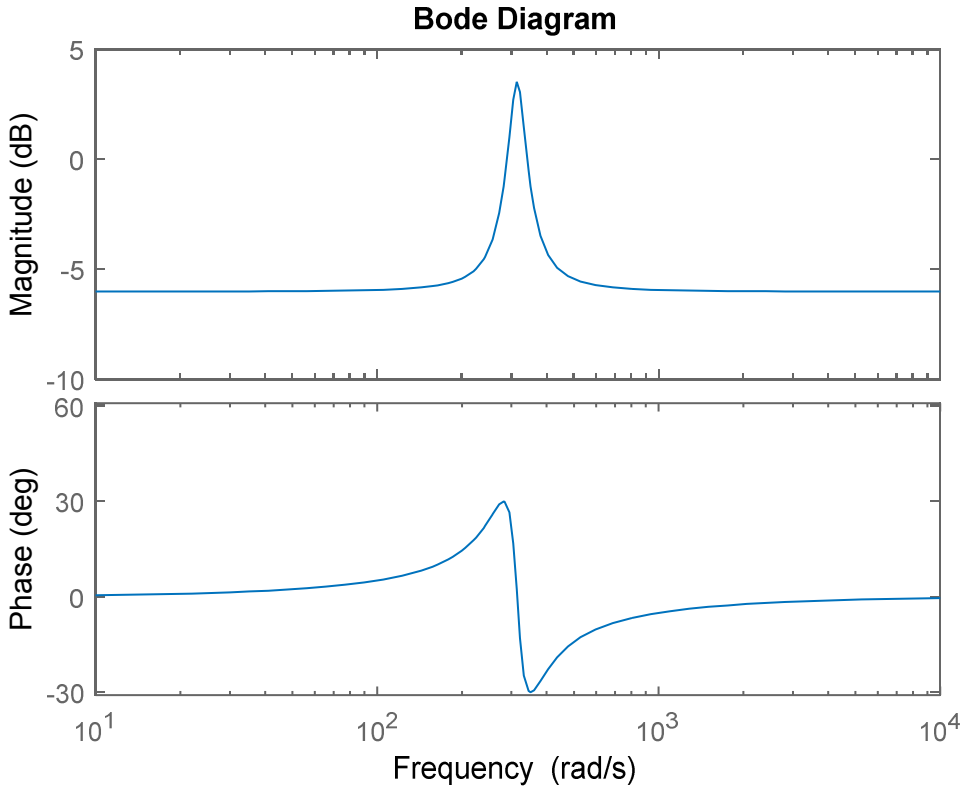


Figure 5.2: non-ideal PR compensator

Another preference of 5.13 is that, unlike 5.12 by adjusting $\omega_c$ properly, the sensitivity towards slight frequency variation could be reduced while the bandwidth is widened. Smaller $\omega_c$ makes more selective transfer function and the peak of resonant controller is narrower. However, the smaller $\omega_c$ would make the filter more sensitive to frequency variations, thus the transient response is slower. In practice, $\omega_c$ values 5-15 [rad/s] are a good compromise.

In the paper '' Rotating Transformation and Resonant Control based Feedback Control Strategy for Dynamic Voltage Restorer System'' of Suxuan Guo, Dichen Liu, Member, IEEE [6] is proposed a control technique which is a frequency selective technique aims at compensating specific harmonic frequencies. The error signal is transformed from the stationary frame to the rotating frame, and then induced into the feedback control block based on the proportional resonant control. For the fundamental component, a proportional is tuned on the fundamental frequency and it is transformed into the DC component using a PI controller; for the couple of harmonics at $(6k \pm 1)\omega_1$ harmonic frequencies, a single proportional resonant controller is tuned at the frequency $6k\omega_1$, two proportional resonant controllers in the stationary frame could be reduced to one controller in the rotating frame.

### *5.4 Test in a continuous domain*

Finding the theory of the PR in the papers is not an easy task because in the papers it is rare that the theory is explained completely. Thus, we have done a test in continuous to verify if they work well before to implement them for our control in a discrete domain.

To prove if they work, we compared the errors between a closed loop with only PI and one with also PRs in parallel, introducing a 7$^{th}$ and a 13$^{th}$ harmonics in abc reference and a constant error. The 7$^{th}$ and the 13$^{th}$ harmonics became a 6$^{th}$ and a 12$^{th}$ in a rotating frame, so the regulator made by the two PR and the integrator is showed in figure 5.3 and its transfers function is represented by the bode diagrams in figure 5.4 like follows.

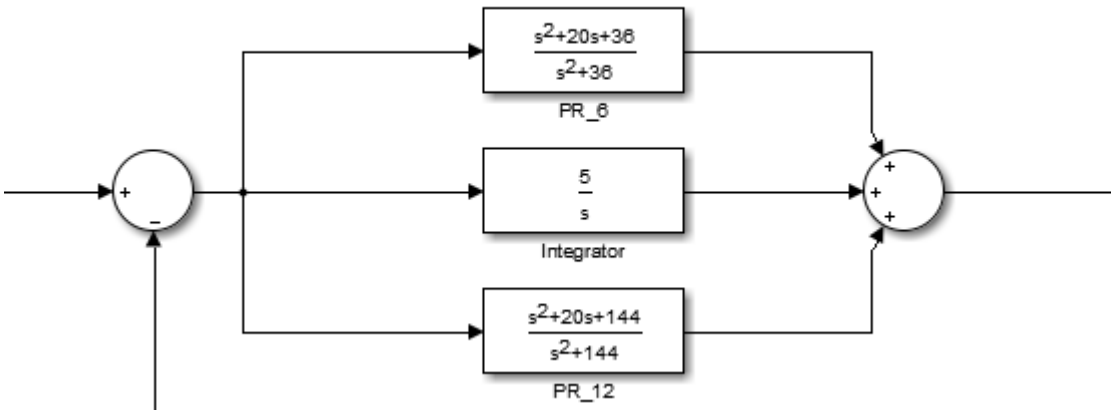


Figure 5.3

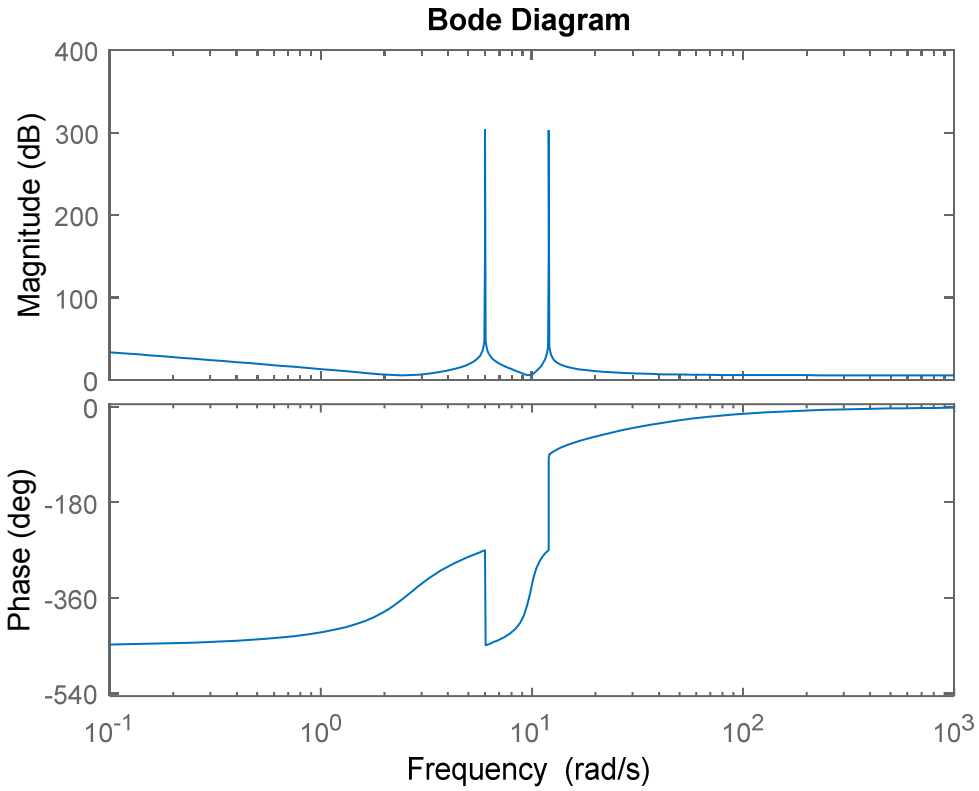


Figure 5.4

Now let's show the results obtained comparing the two different regulators. In figure 5.5 we can see the two errors compared and it's clear that the regulator with the PRs is much better than the other one which can't compensate the 6$^{th}$ and the 12$^{th}$ harmonics effect.

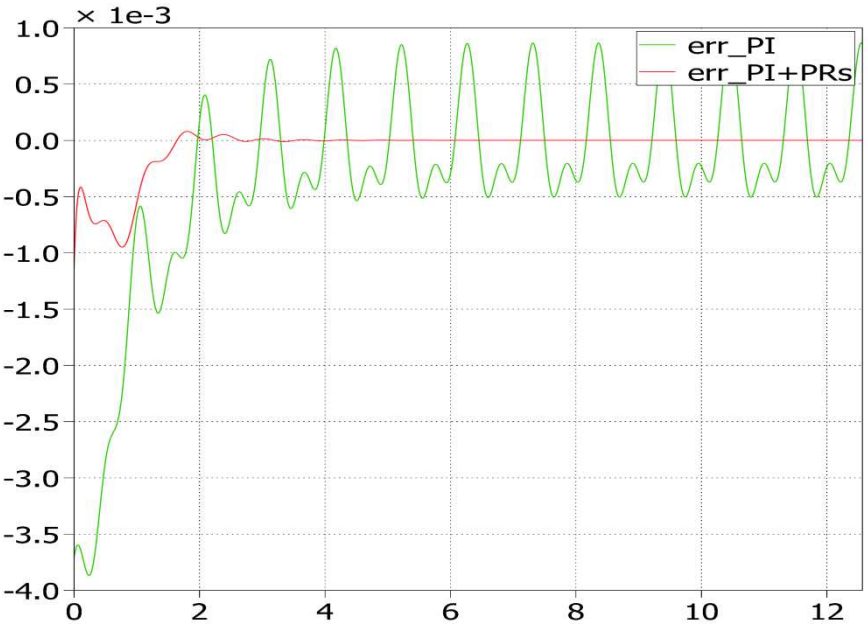


Figure 5.5

The harmonic spectrum in the last fundamental period ($2\pi$) of figure 5.5 is displayed in figure 5.6.

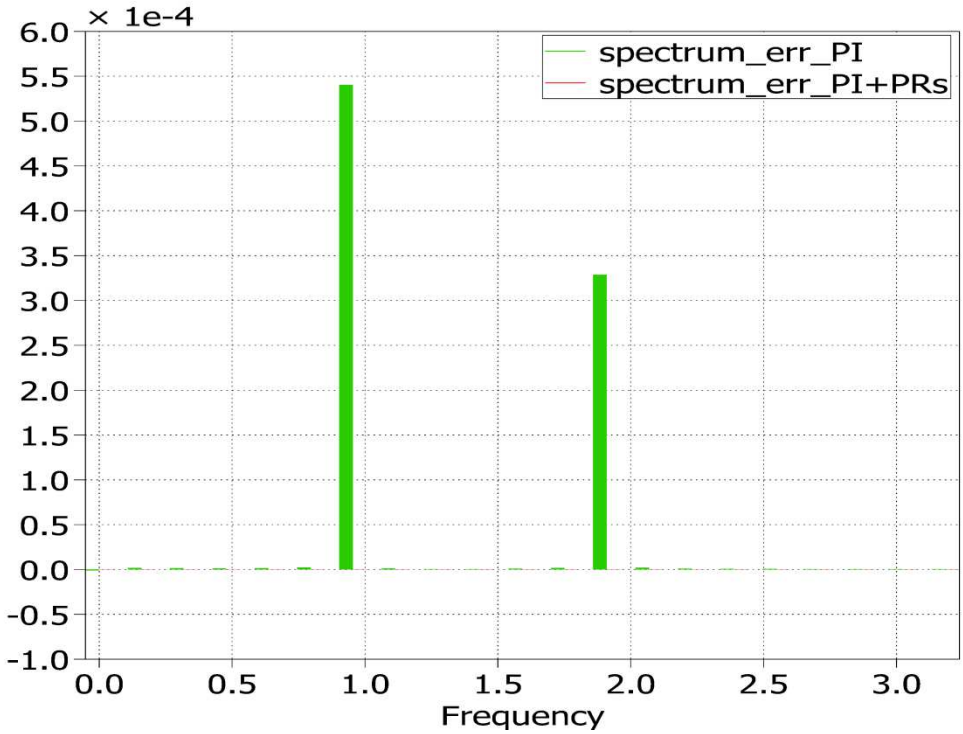


Figure 5.6

Like we can observe using PRs, we have neglected the 6$^{th}$ and the 12$^{th}$ harmonics.

In figures 5.7 and 5.8 are showed the reference and measured signals using a simple PI and using also PRs in parallel respectively.

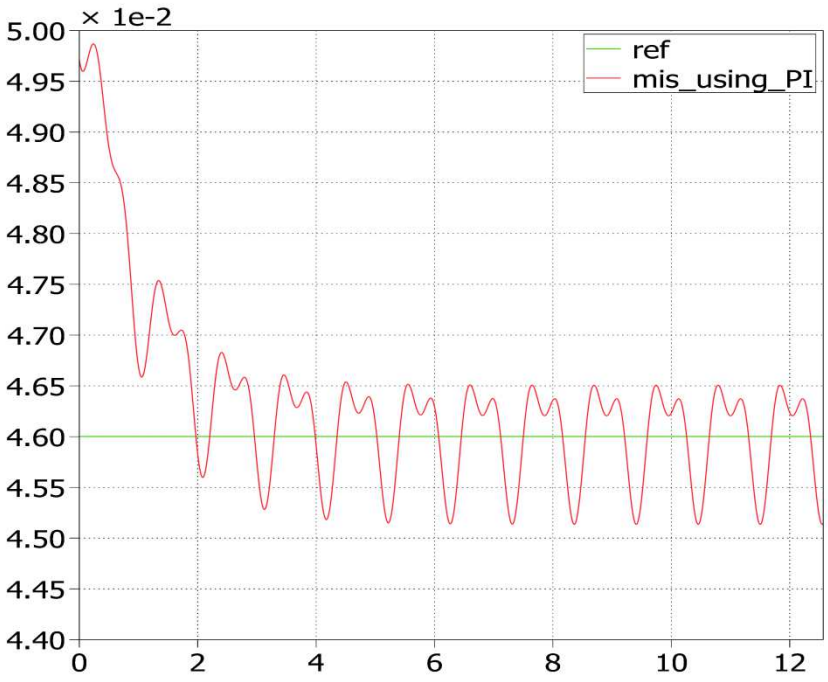


Figure 5.7

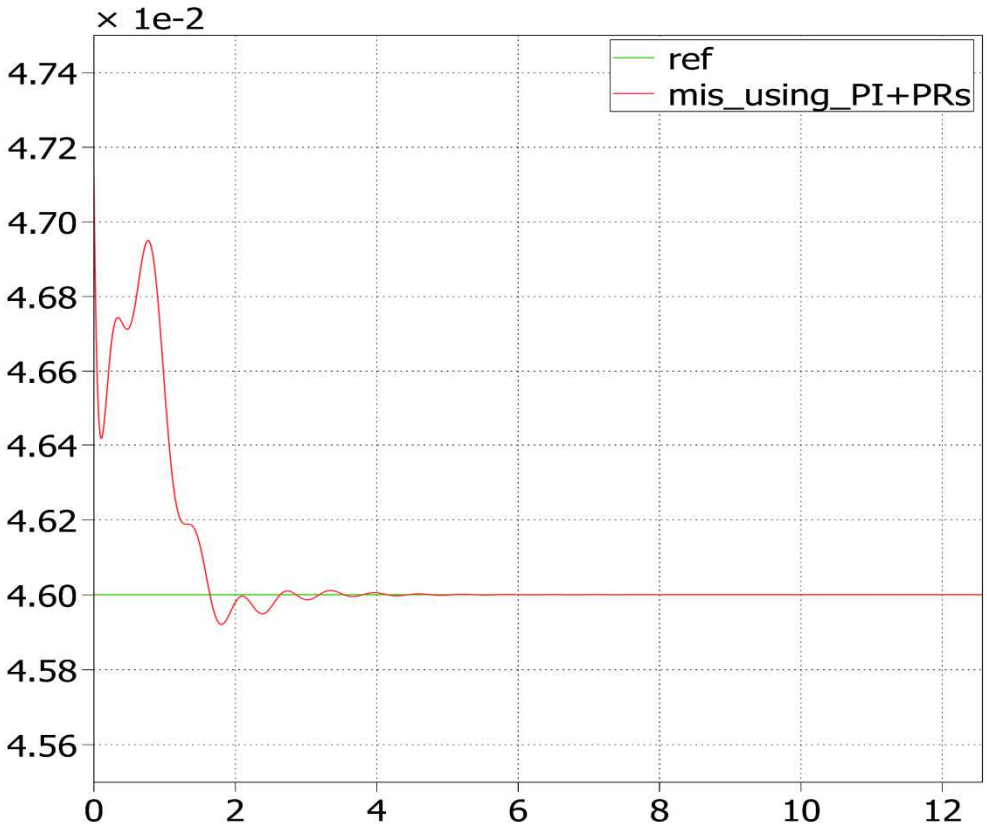


Figure 5.8

Now that we are sure that PRs work, we have implemented them in a discrete domain only to neglect the 6$^{th}$ harmonic introduced by the dead time in a rotating frame.

### *5.5 Discretization*

For the discretization of the PRs we have used the bilinear transform, also known as Tustin's method, which is a first-order approximation of the natural logarithm function that is an exact mapping of the z-plane to the s-plane. When the Laplace transform is performed on a discrete-time signal (with each element of the discrete-time sequence attached to a correspondingly delayed unit impulse), the result is precisely the Z transform of the discrete-time sequence with the substitution of:

$$z = e^{sT} \qquad (5.14)$$

$$e^{sT} = \frac{e^{\frac{sT}{2}}}{e^{-\frac{sT}{2}}} \qquad (5.15)$$

And using Taylor until the first order we have:

$$z = \frac{1 + s\frac{T}{2}}{1 - s\frac{T}{2}} \qquad (5.16)$$

where $T$ is the numerical integration step size of the trapezoidal rule used in the bilinear transform derivation, or in other words, the sampling period. The inverse of this mapping (and its first-order bilinear approximation) is:

$$s = \frac{1}{T}\ln(z) \qquad (5.17)$$

Using again Taylor until the first order of approximation we obtain

$$s = \frac{2}{T}\frac{z-1}{z+1} \qquad (5.18)$$

The Tustin or bilinear approximation yields the best frequency-domain match between the continuous-time and discretized systems.

In our case it's important to have a good dynamic at the resonant frequency, well we have used the Tustin method with frequency prewarping. This method ensures a match between the continuous- and discrete-time responses at the prewarp frequency. The Tustin approximation with frequency prewarping uses the following transformation of variables:

$$s = \frac{\omega}{\tan\left(\frac{\omega T}{2}\right)}\frac{z-1}{z+1} \qquad (5.19)$$

This change of variable ensures the matching of the continuous- and discrete-time frequency responses at the prewarp frequency ω, as we can observe by the following demonstration.

$$z = e^{j\omega T} \qquad (5.20)$$

If we substitute the 5.20 in 5.19 we have:

$$s = \frac{\omega}{\tan\left(\frac{\omega T}{2}\right)}\frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} \qquad (5.21)$$

$$= \frac{\omega}{\tan\left(\frac{\omega T}{2}\right)}\frac{e^{\frac{j\omega T}{2}}\left(e^{\frac{j\omega T}{2}} - e^{-\frac{j\omega T}{2}}\right)}{e^{\frac{j\omega T}{2}}\left(e^{\frac{j\omega T}{2}} + e^{-\frac{j\omega T}{2}}\right)} \qquad (5.22)$$

$$= \frac{\omega}{\tan\left(\frac{\omega T}{2}\right)}\frac{\left(e^{\frac{j\omega T}{2}} - e^{-\frac{j\omega T}{2}}\right)}{\left(e^{\frac{j\omega T}{2}} + e^{-\frac{j\omega T}{2}}\right)}\frac{2}{2j}j \qquad (5.23)$$

$$= \frac{j\omega}{\tan\left(\frac{\omega T}{2}\right)} \frac{\frac{\left(e^{\frac{j\omega T}{2}} - e^{-\frac{j\omega T}{2}}\right)}{2j}}{\frac{\left(e^{\frac{j\omega T}{2}} + e^{-\frac{j\omega T}{2}}\right)}{2}} \qquad (5.24)$$

Using Eulero formulas we have:

$$= \frac{j\omega}{\tan\left(\frac{\omega T}{2}\right)} \frac{\sin\left(\frac{\omega T}{2}\right)}{\cos\left(\frac{\omega T}{2}\right)} \qquad (5.25)$$

$$= \frac{j\omega}{\tan\left(\frac{\omega T}{2}\right)} \tan\left(\frac{\omega T}{2}\right) \qquad (5.26)$$

So finally

$$s = j\omega \qquad (5.27)$$

Now let's try to discretize a PR transfer function like the 5.13.

```
close all
clear all
clc

%%% esempio di di scretizzazione di una fdt di un PR mettendo a confronto
Tustin e Tustin + frequenza di pre-warmap

% definiamo 's'

s=tf('s');
C=0.5+20*s/(s^2+20*s+100^2);

%% Tustin

bodeplot(C)
Cdz = c2d(C,0.01,'tustin');
bodeplot(C,Cdz)
legend('C','Cdz');

%% Tustin + fwp

discopts = c2dOptions('Method','tustin','PrewarpFrequency',3.0);
Cdzp = c2d(C,0.001,discopts);
bodeplot(C,Cdz,Cdzp)
legend('C','Cdz','Cdzp')
```

In the following figure we can observe the efficacy of the ’PrewarpFrequency’.

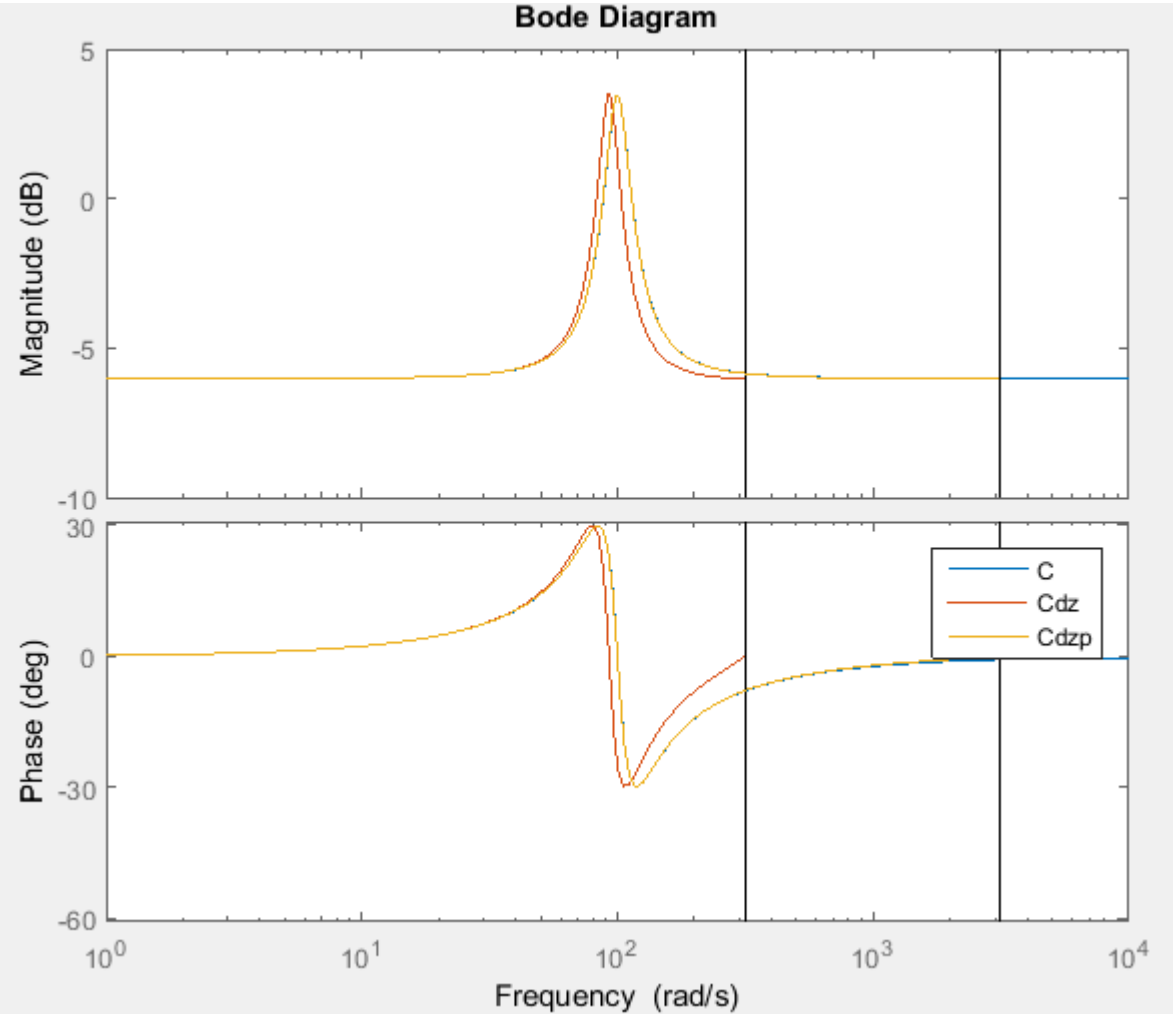


Figure 5.9

# Chapter 6:

# Simulation final results

### *6.1 Introduction*

In this final chapter we can observe the final results obtained by the simulation. We will see that the requirements of the FACRI project for the part of control are been satisfied. The final scheme used for the control is made of all the algorithms described in the previous chapters and now we will show their results.

### *6.2 Requirements*

Thanks a lookup table we make the requirements of speed and torque. To satisfy them, we have to be able to give that minimum torque for the wanted speed and they are represented in the following figure.

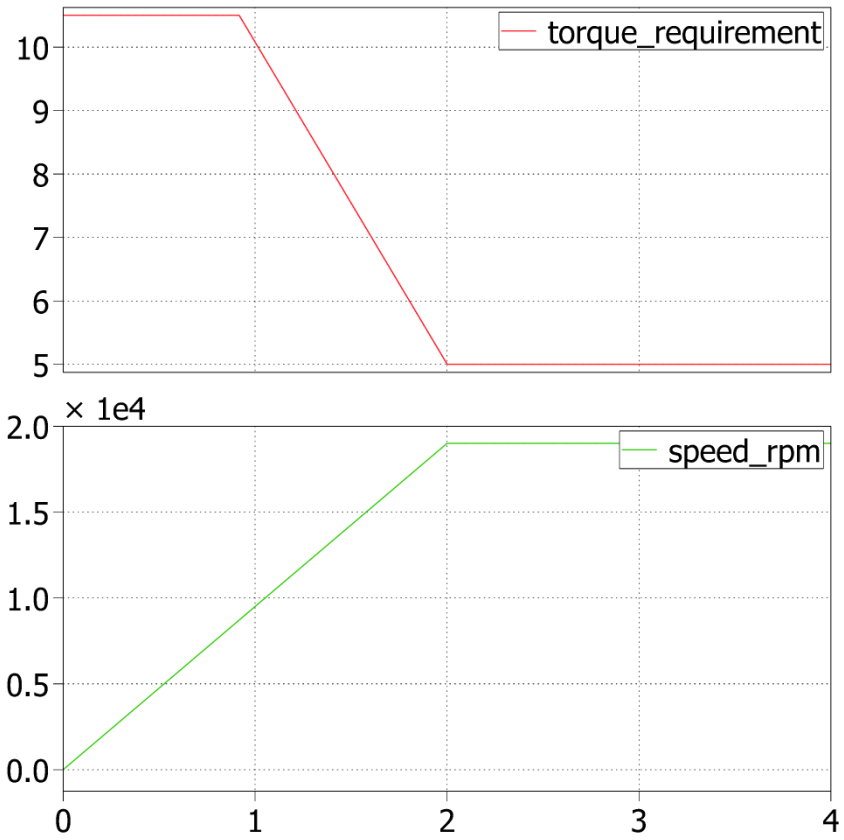


Figure 6.1: requirements

Thus, we have to produce a torque of 10.5 [Nm] until 9000 [rpm] and 5 [Nm] at full speed 19000 [rpm], like we can observe in figure 6.1.

### *6.3 MTPA and Flux Weakening*

The inputs necessary to make the reference currents $i_d$ and $i_q$ are the mechanical-electrical speed reference $\omega_{me}$ and the maximum torque wanted $m = 10.5\ [Nm]$. In figure 6.2 is showed the Simulink scheme.

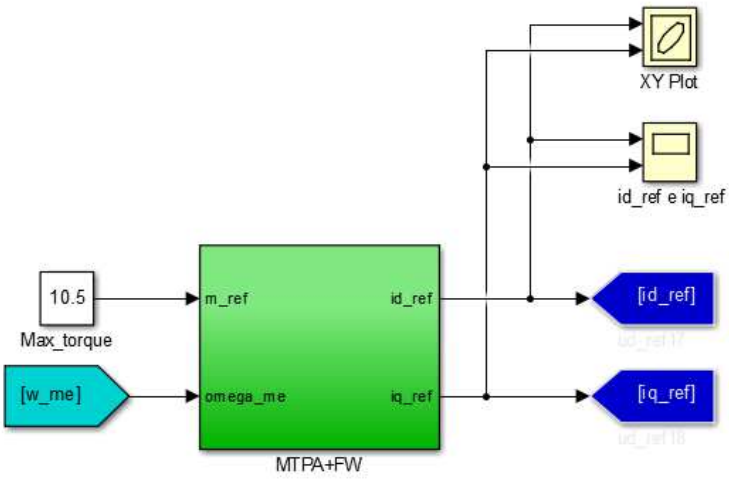


Figure 6.2

In the chapter 2, we have seen that in the upper green block are calculated the MTPA references before and the FW references later, so in figure 6.3 are showed the MTPA references and in figure 6.4 the final currents in a XY plot.

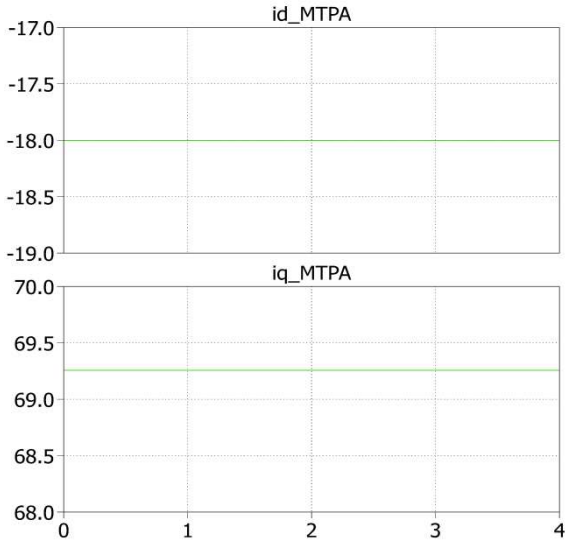


Figure 6.3: MTPA references

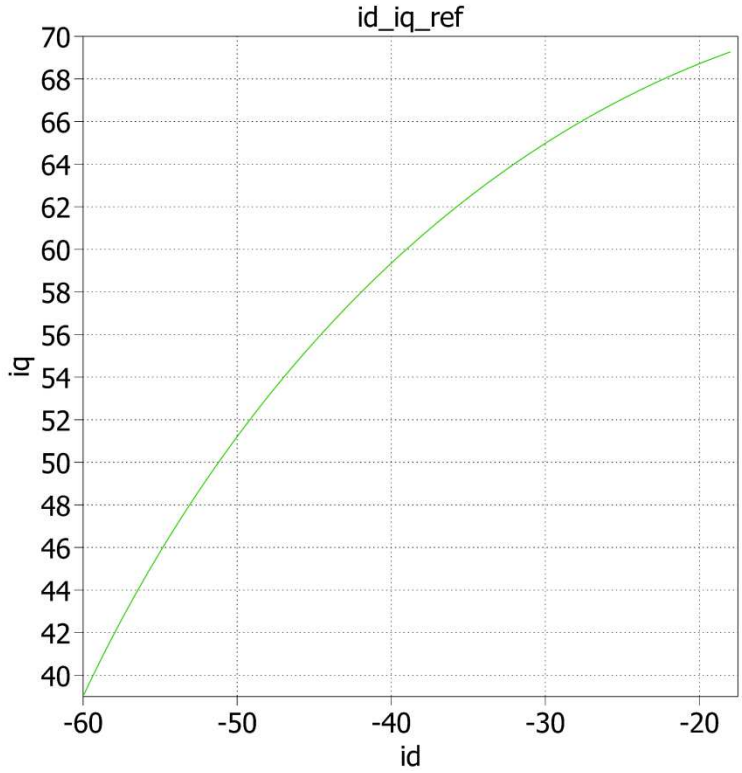


Figure 6.4: I_dq references

Now let's show the real currents measured in the output of the motor.

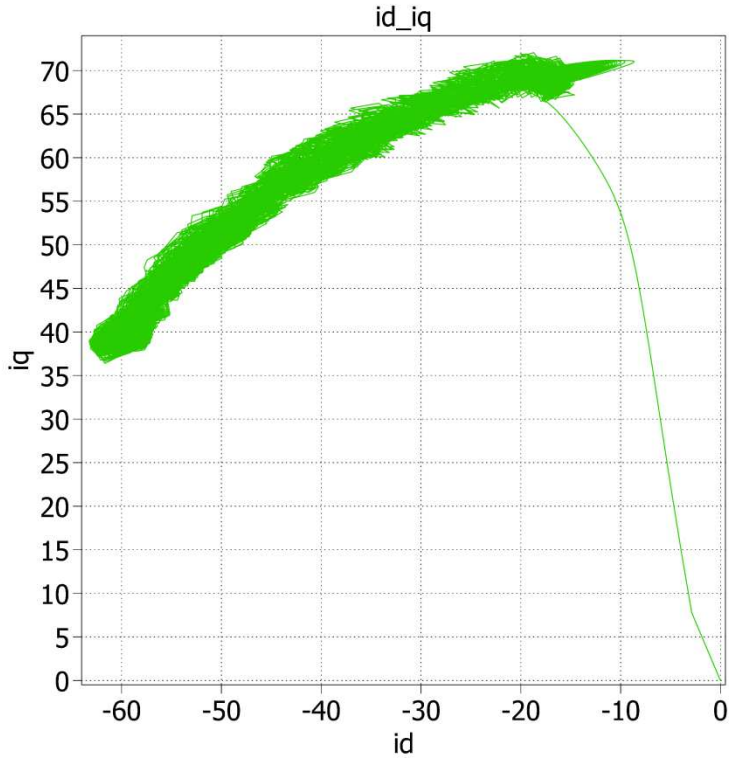


Figure 6.5: I_dq real

### *6.4 Voltage references*

The voltage references are calculated by the PI and PR regulators form the errors between the reference currents and reals, like showed in the figure 6.6.

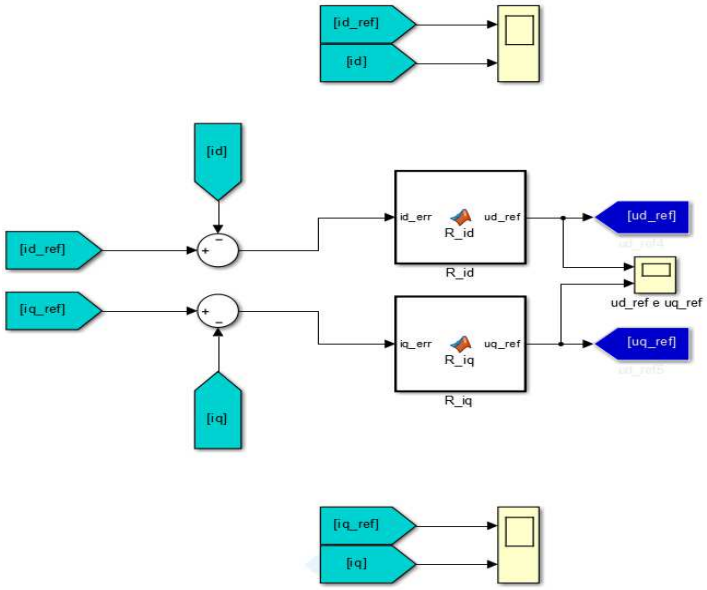


Figure 6.6

Now let's see the references with the measures and the voltage outputs in the figures 6.7, 6.8, and 6.9 respectively.

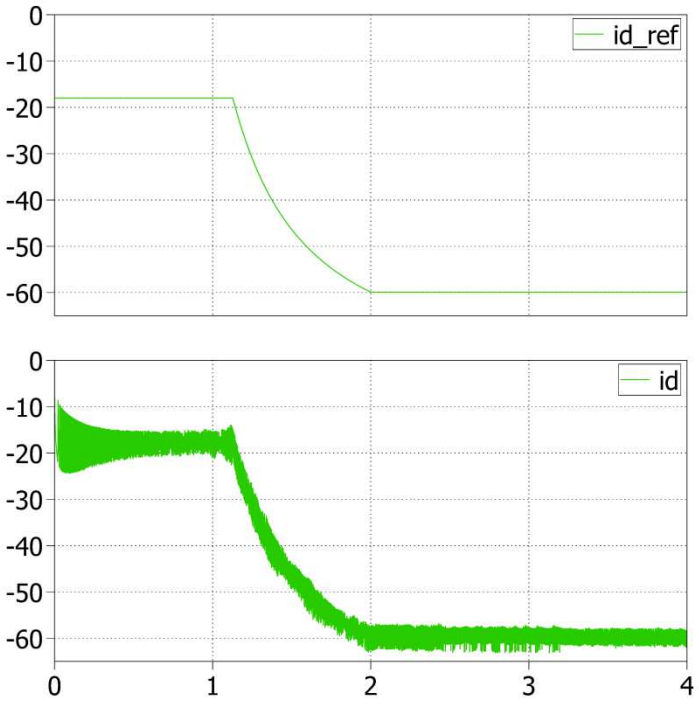


Figure 6.7: id_ref and id

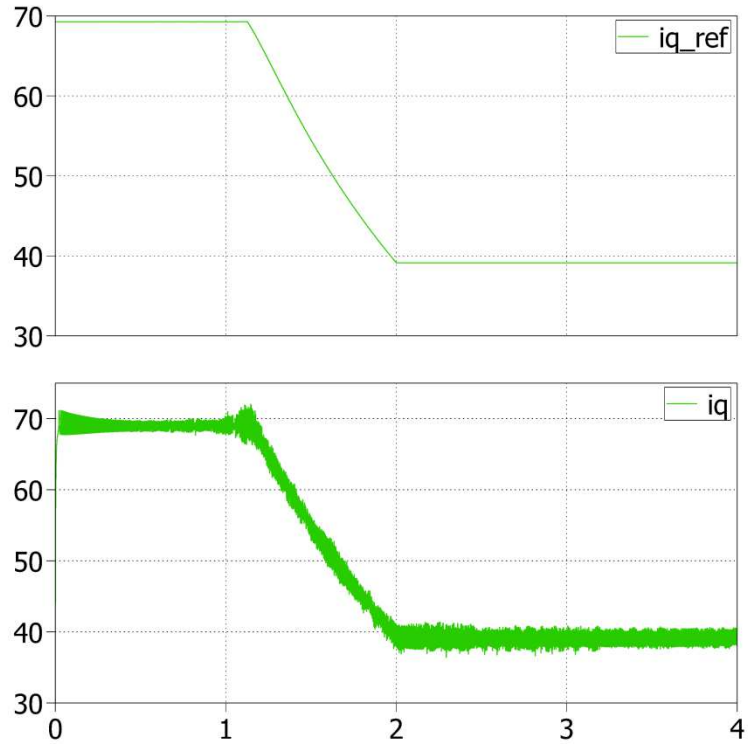


Figure 6.8: iq_ref and iq

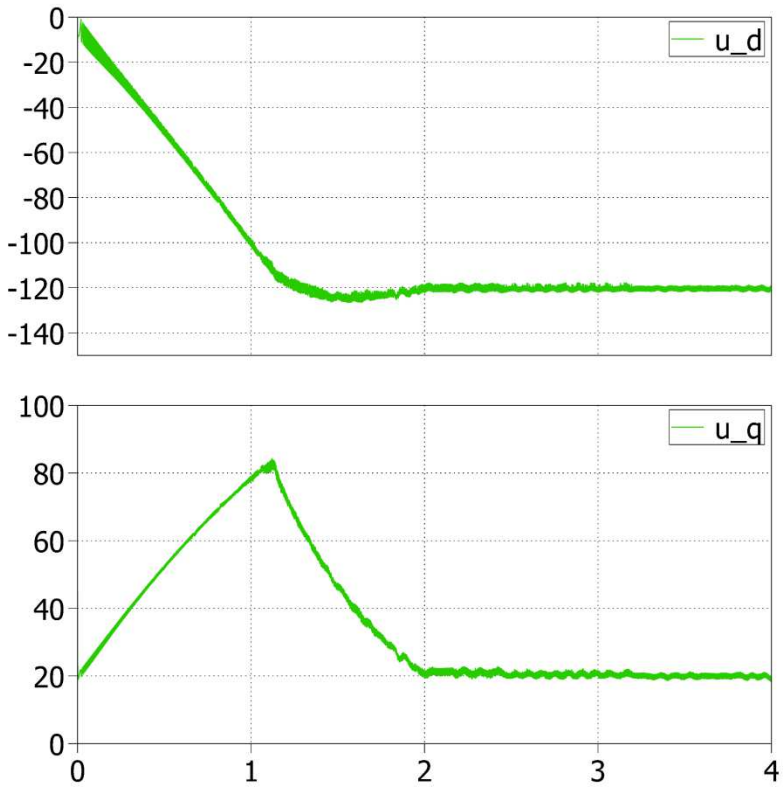


Figure 6.9: voltage references

### *6.5 Overmodulation*

The voltage references are been calculated without any type of voltage limiter, thus we can apply the algorithm described in the chapter 4 to do overmodulation if necessary. In the following figure is reported the subsystem block where there is inside the algorithm.

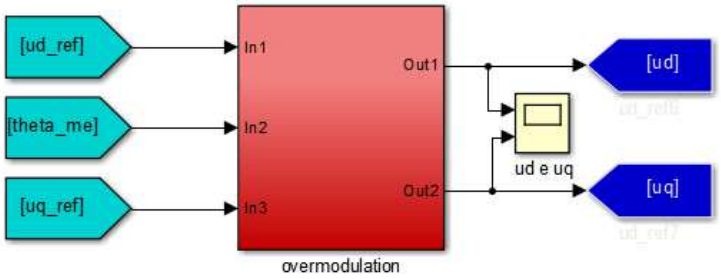


Figure 6.10

The outputs are normalized with half DC-link and transformed in a,b,c coordinates to became the inputs of the modulator. Now let's observe in figure 6.11 the carrier with the normalized voltage of the phase a.

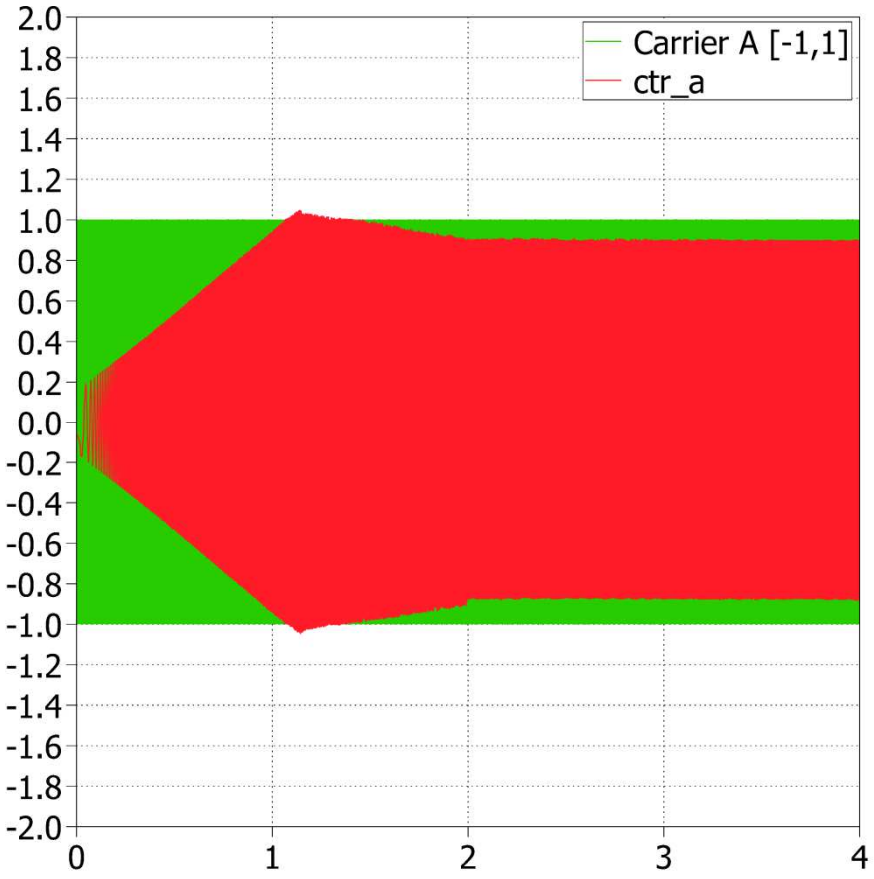


Figure 6.11

In figures 6.12 and 6.13 are reported the enlargements of the overmodulation zone in the upper figure and the full speed zone respectively.

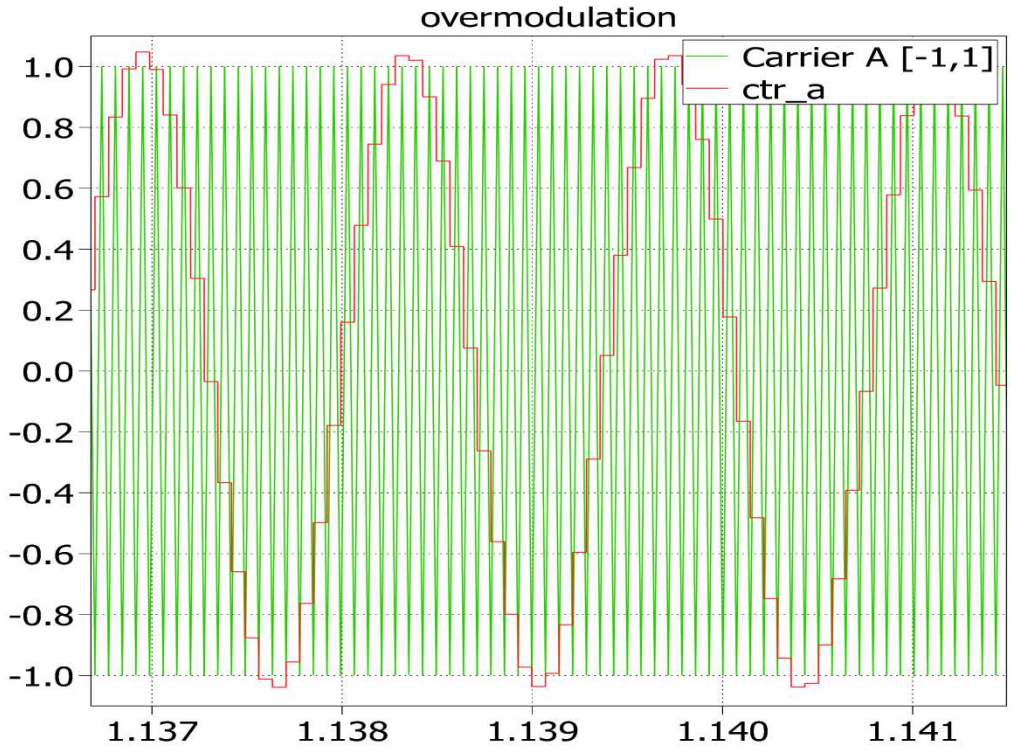


Figure 6.12


Figure 6.13

### *6.6 Final results*

In this paragraph are showed the final results that are the most important. In figure 6.14 and 6.15 are displayed the DC-link power and the mechanical power respectively.

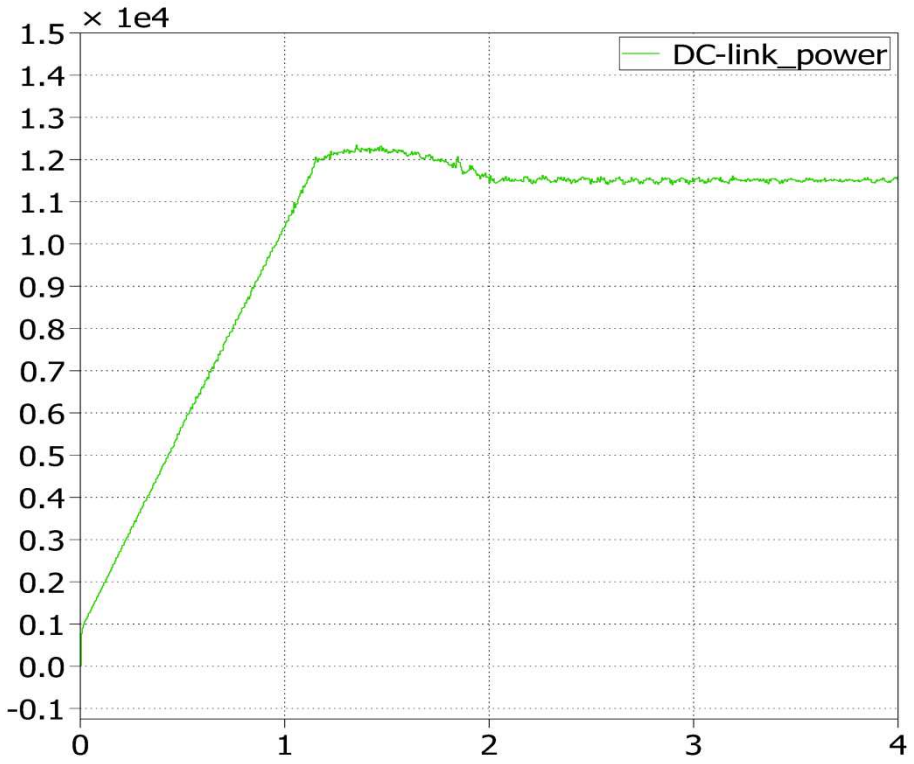


Figure 6.14

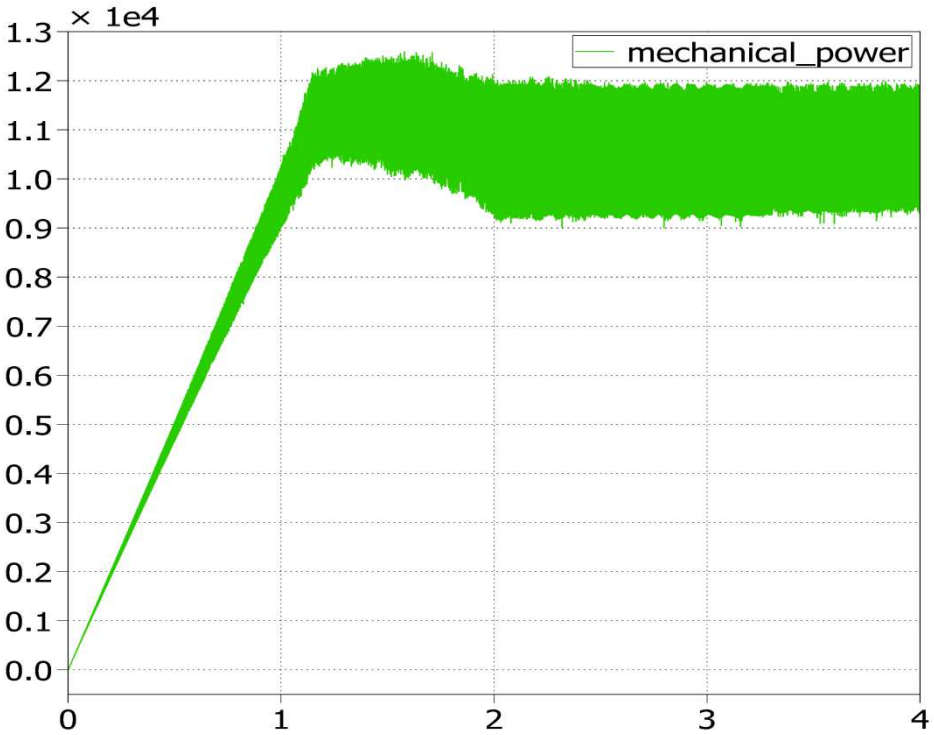


Figure 6.15

Now let's see if the requirements are been satisfied confronting them with the torque and the speed measured.

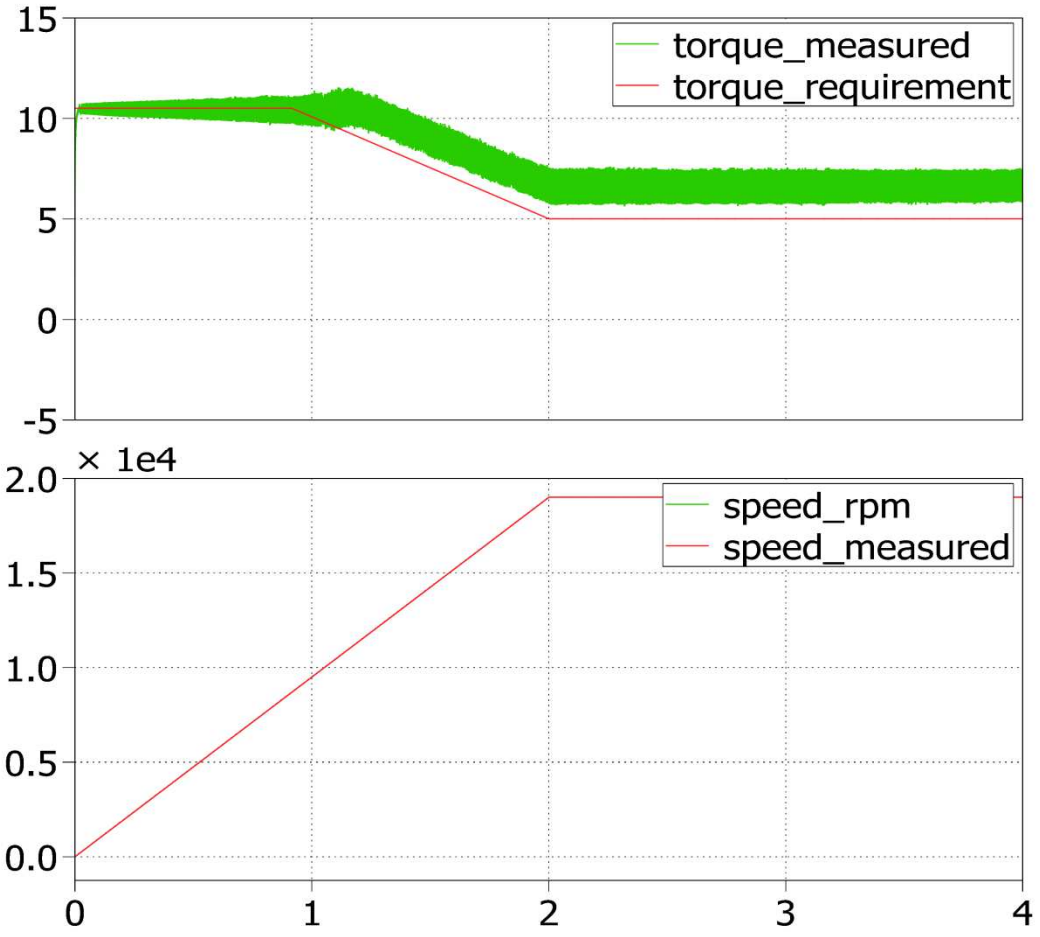


Figure 6.16

It's important to observe that using the overmodulation we can stay more time in the MTPA condition because we can use more DC-link, thus the voltage limit is bigger and the FW is reached later.

The phase currents are displayed below during the MTPA and during full speed conditions with the respectively harmonic spectrums.

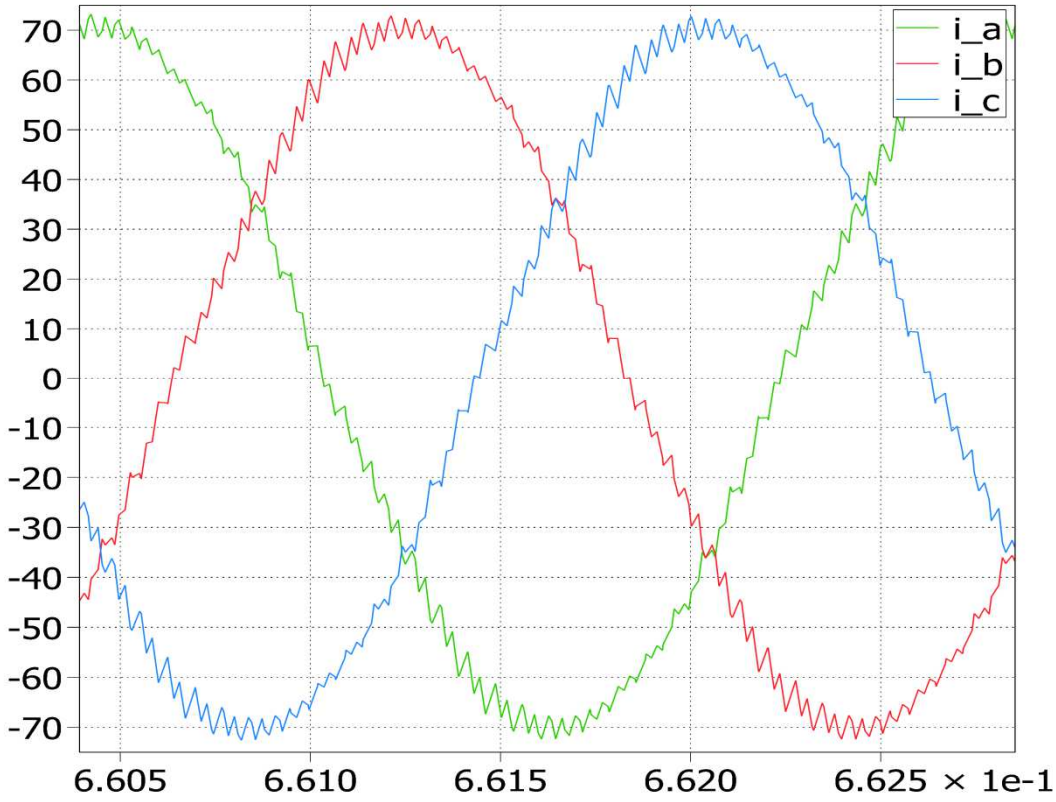


Figure 6.17: I_abc in MTPA condition

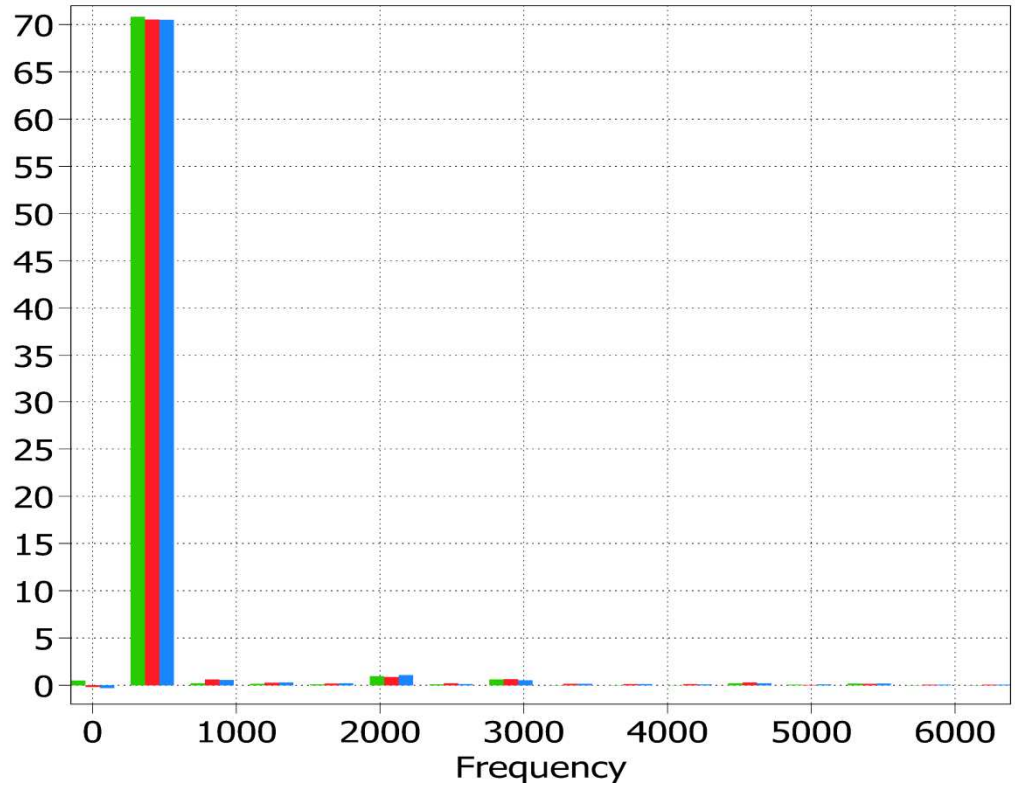


Figure 6.18: harmonic spectrum of I_abc_MTPA

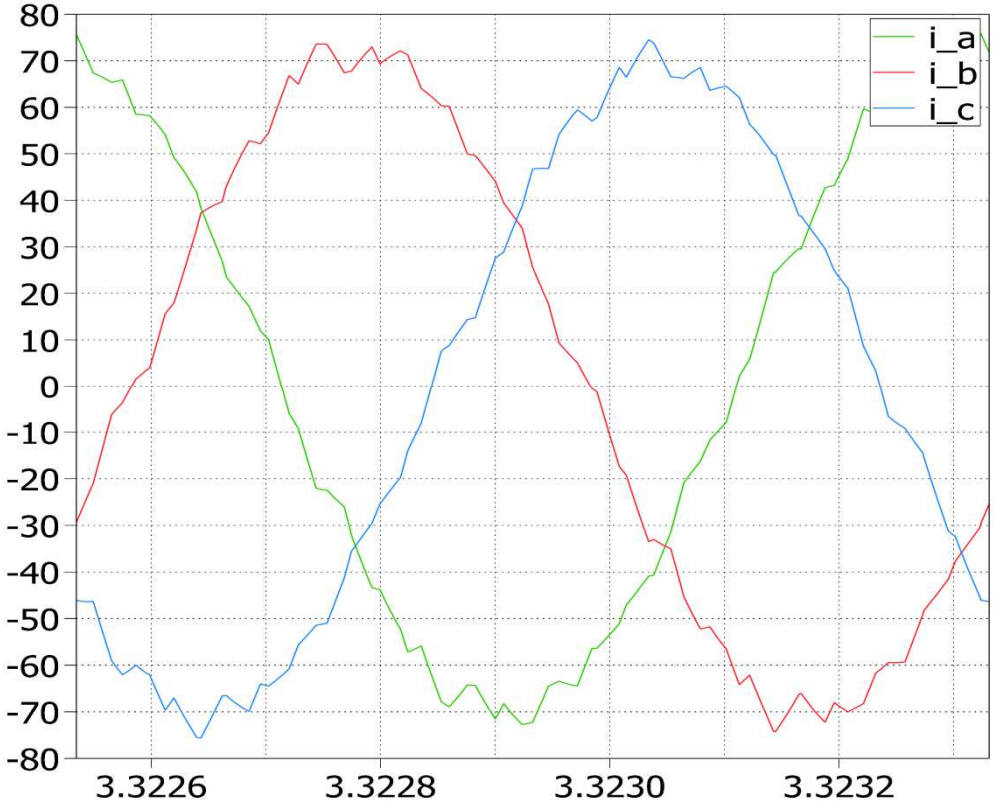


Figure 6.19: I_abc at full speed

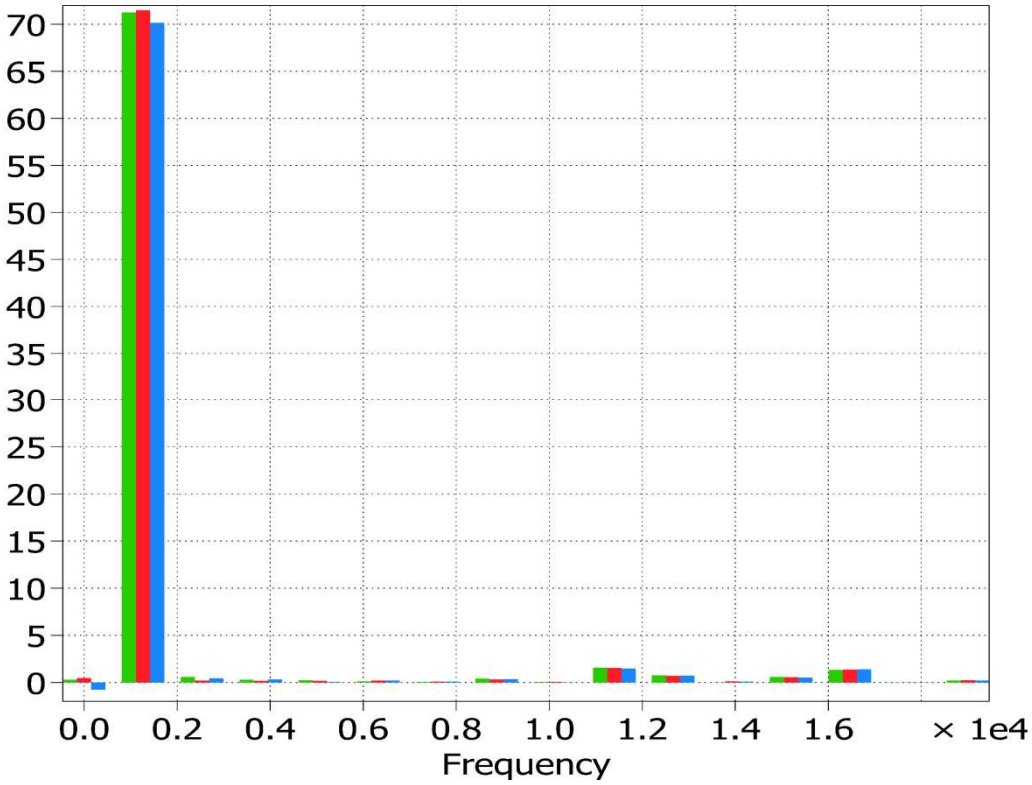


Figure 6.20: harmonic spectrum of I_abc_full_speed

In figure 6.20 we can observe that there are the $9^{th}$ and the $13^{th}$ harmonics that are caused by the switching frequency equal to 13933.3 [Hz] like previously demonstrated in figure 3.15 of chapter 3.

# Chapter 7

# Real components

### *7.1 Introduction*

In the trade-off study different Power Electronics Converter (PEC) topologies have been compared with the aim of identifying, according to the project specifications, the most suitable configuration for the actual application. As already discussed, each PEC topology provides pros and cons, and a qualitative comparison has been provided to highlight the differences. According to the performed review and the conclusions achieved in it, the PEC topology chosen for the project is a 2-Levels full-bridge configuration based on IGBTs power modules, whose schematic structure is depicted in **Errore. L'origine riferimento non è stata trovata.**.

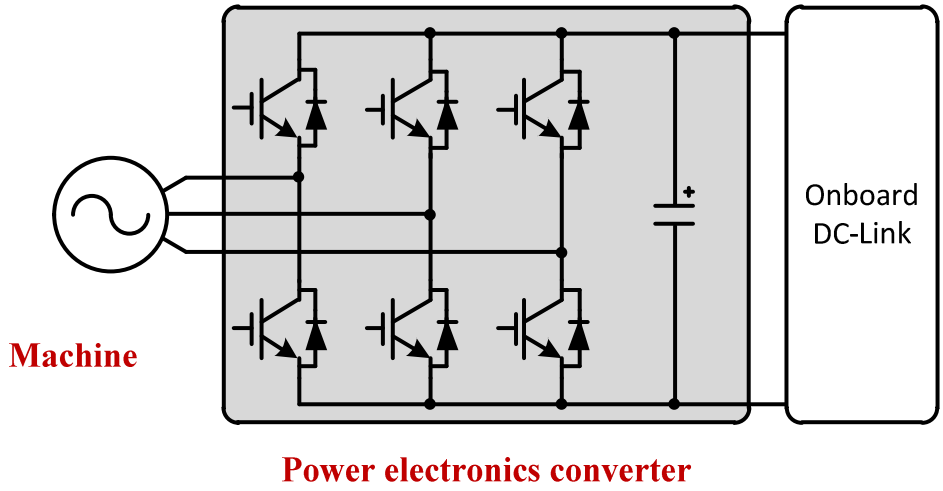


Figure 7.1: Power Electronics Converter unit schematic structure

### *7.2 Power electronic converter design*

Some initial requirements and constraints have been provided by FACRI for the PEC unit during the proceedings of the project. In this part we tried to collect and report them for the sake of completeness.

Supply voltage

- Main supply voltage (for PEC): 270V (235V-310V)
- Secondary power supply (for control platform and auxiliaries): 28V (18V-36V)
- Physical isolation between power supplies: 3000Vpk

Output current

- Maximum output current: 85 A

PEC protections

- DC-Link pre-charge circuit
- DC-Link Over-voltage shutdown
- DC-link Under-voltage shutdown
- Output phase over-current
- PEC power module over-temperature shutdown
- Machine over-temperature shutdown

Machine position sensor:

- Machine position sensor: sin-cos resolver.

Dimensions:

- Overall dimensions: 190mm*210mm*150mm (excluding connectors and supports)
- Weight: ≦ 5kg

Operating temperatures and cooling:

- Operating temperatures: -40°C / +70°C
- Cooling method: forced air cooling

Based on the converter output peak current (85A), the maximum operating temperature (70°C), the cooling method (forced air cooling), the DC-Link voltage (270V) and the selected devices technology (IGBTs), the core of PEC unit has been selected to be the Infineon IGBT power module FS200R07N3E4R. This choice has come after considering different power modules available on the market able to suit in the specific application and after considerations in terms of availability (lead times, costs, suppliers) and size (FACRI provided some stringent requirements in terms of size of the PEC unit, so the potential more compact structure has been selected). The selected power module has a maximum continuous current of 200A, a maximum reverse voltage of 650V and comes in a six pack arrangement, allowing to keep the overall design as compact as possible. The 650V maximum reverse voltage has been selected because of the reduced operating DC-Link voltage: using a 1.2kV module would have just led to higher switching and conduction losses (and thus to increase cooling requirements) without providing any additional benefit. **Errore. L'origine riferimento non è stata trovata.** and **Errore. L'origine riferimento non è stata trovata.** show the actual aspect of the selected power module and its internal structure.

Figure 7.2: power module actual aspect

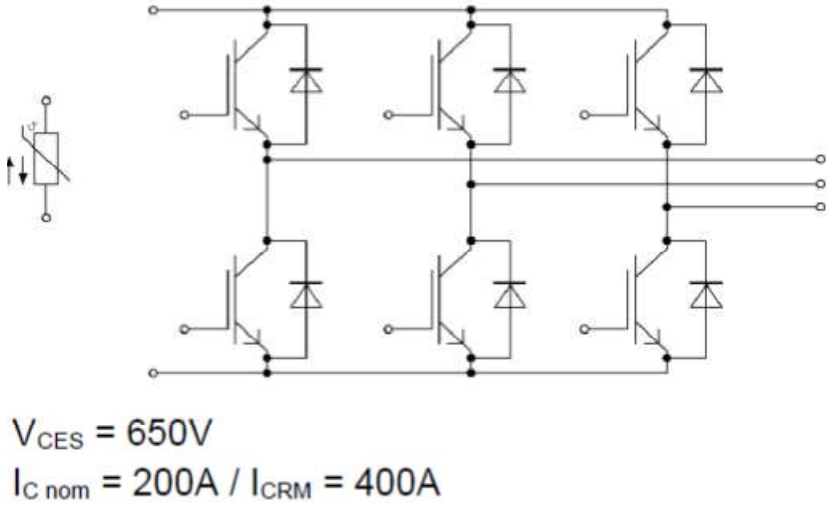


Figure 7.3: power module internal structure

The FS200R07N3E4R power module has been simulated using the programs Simulink and PLECS to evaluate conduction and switching losses. Achieved simulation results present a total PEC losses of about 210W (60% switching, 40% conduction).

Starting from the estimated power losses value (that can be considered valid with a relatively good degree of precision) a suitable heatsink has been selected using the online tool “R-Tools” available from the Canadian company Mersen. It provides for a basic thermal FE analysis where boundaries condition as ambient temperature, as well as source power and air flow can be selected. For the actual application, the maximum ambient temperature of 70°C has been set by FACRI and, in the absence of specifications about the available air flow, the value of 100CFM has been selected. The so obtained cooling system is depicted in figure 7.4.

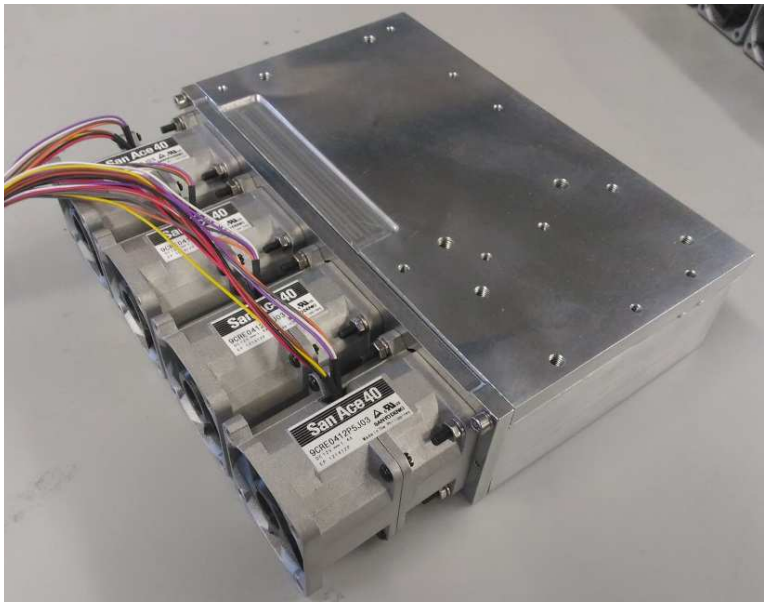


Figure 7.4:  Actual cooling system for the PEC unit

Finally, figure 7.5and **Errore. L'origine riferimento non è stata trovata.** show the actual prototype at the University of Nottingham, without and with the DC-Link capacitor board. In those figures, the final arrangement of the power layer and gate drivers can be seen, as well as the current sensors and their output connector.

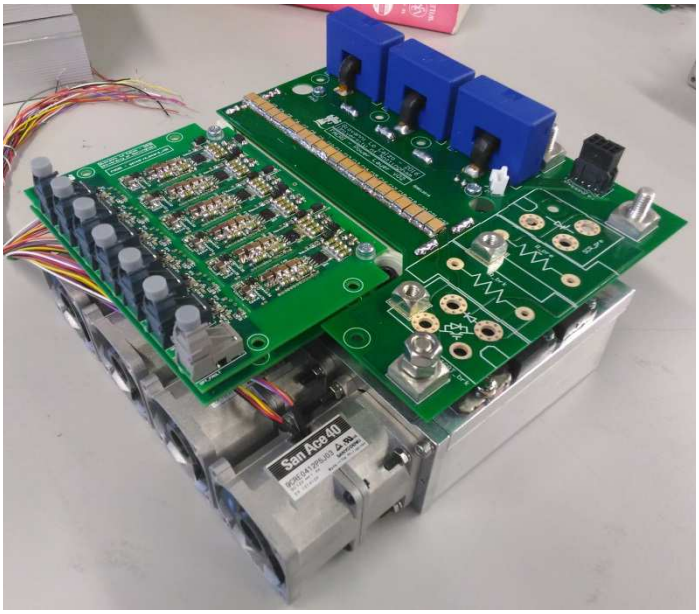


Figure 7.5:  PEC unit without DC-Link capacitors

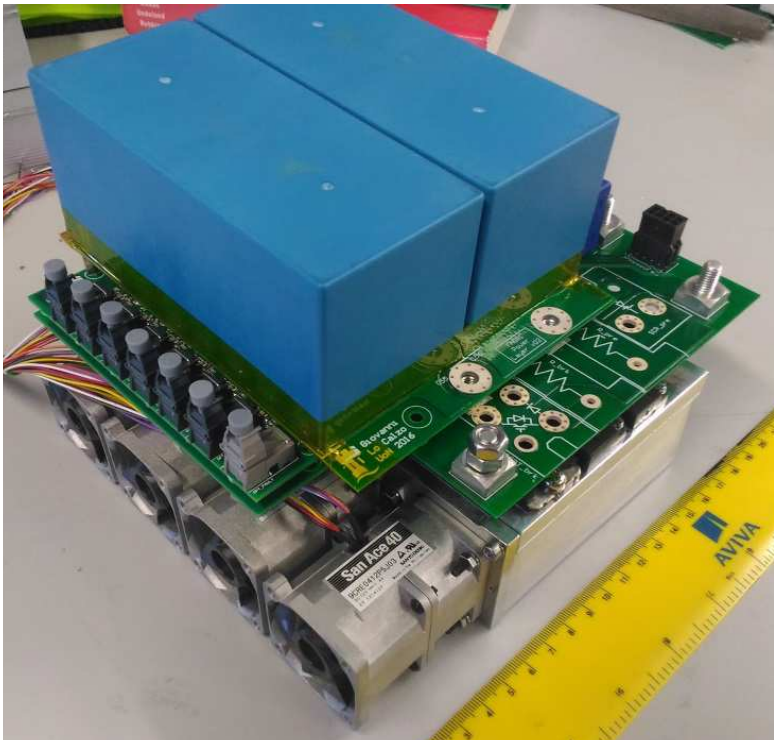


Figure 7.6: PEC unit with DC-Link capacitors

### *7.3 DSP*

A digital signal processor (DSP) is a specialized microprocessor with its architecture optimized for the operational needs of digital signal processing. The controller platform for drives and converters used in the lab has the followings characteristics:

- 24x Fibre optics links
- Intrinsically isolated
- Each link configurable as output (power devices driving) or input (faults reading from power layer or system)
- Power supplies for the FPGA fabric banks
- 16x ADC channels with 14bit resolution and 2.5Msps sampling capability
- Antialiasing pre-filters
- Possibility to use oversampling to increase the number of bits (depending on the application switching frequency)
- Each channel configurable as unipolar or bipolar channel in order to fully exploit ADC range with e.g. DC-link voltage sensors
- High speed analogic comparators and digitally programmable voltage dividers for fast detection of hardware faults (response time limited by sensors bandwidth)
- Comparators output directly connected to FPGA fabric for fast shut-down in the event of

faults

- Carrier buffering and current limiting
- Single ended or differential
- Isolated digital interfaces
- Isolated power supplies

Figure 7.7 shows the DSP projected by Dr. Giovanni Lo Calzo and Dr. Andrea Formentini.

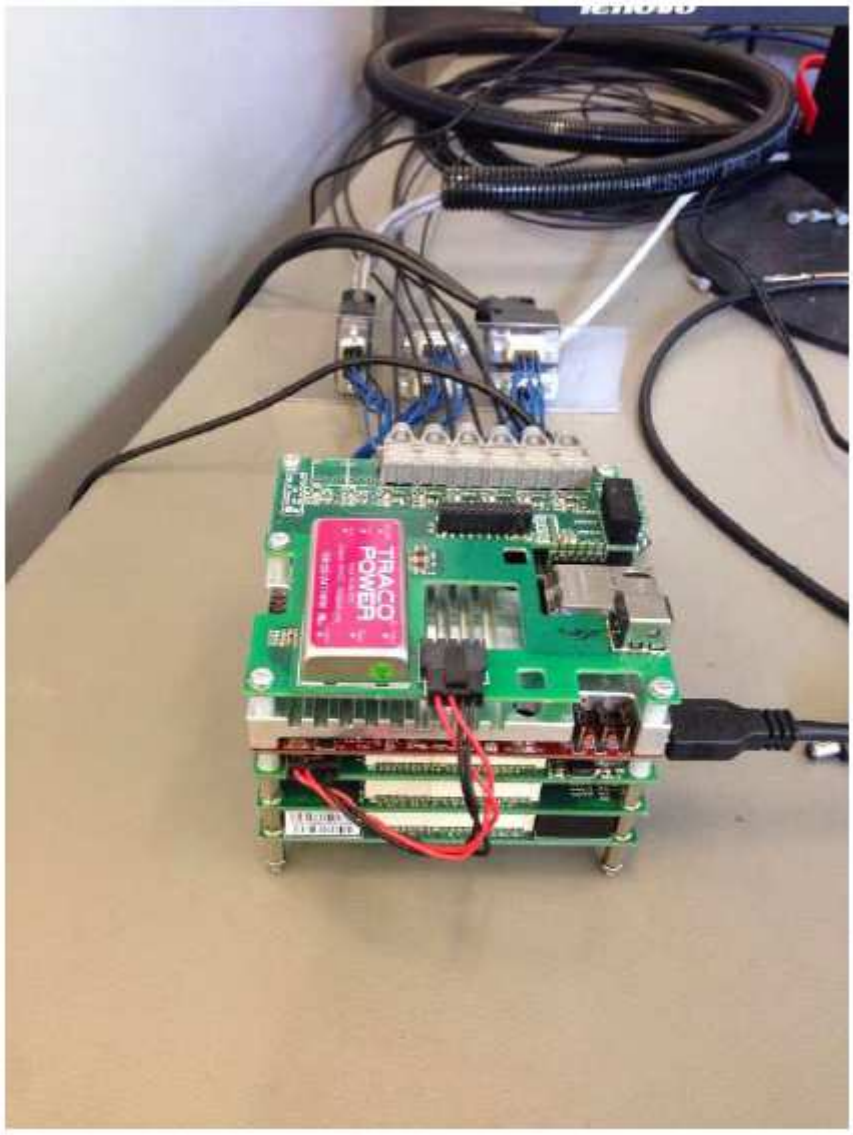


Figure 7.7: DSP

# Conclusion

The field of ''high speed'' is actually always more studied because it has a lot of advantages in terms of power, volume, efficiency, costs etc. In this thesis I have analysed very important topics for the industrial drives. The most important chapter is the 4th where it is explained how is possible passing in overmodulation in a simple and very cheap way from the computational point of view. Using the overmodulation when it is necessary allows us to use better the DC-link and so increase the rated speed. Finally the requirements of the project have been satisfied and the results obtained in simulation seem to be promising. Unfortunately, before to do the experimental tests it will be necessary more time because the motor is still in the planning level and the DSP has to be finished to be implemented. The part of control has been entrusted to me and my supervisor Giovanni Lo Calzo, but we were not alones, in fact there was also an Italian company working on the control of the FACRI motor and we have worked in parallel all the time without any type of contact, finally will be selected the best control obviously. Personally, from this experience in the university of Nottingham I think that I have learned a lot from the technical point of view but not only, actually I have seen also how many differences there are between two country like Italy and UK.

# Bibliography

[1]- ''Novel digital continuous control of SVM inverters in the Overmodulation range'' of Silverio Bolognani and Mauro Zigotto.

[2]- ''Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverter: Part 1-Harmonic elimination'' of Hasmukh S. Patel and Richard G. Hoft.

[3]- '' Using Differential Evolution for Combinatorial Optimization: A General Approach'' of Ricardo S. Prado Rodrigo C. P. Silva and Frederico G. Guimaraes, Oriane M. Neto.

[4]- ''Improved Design and Control of Proportional Resonant Controller for Three-Phase Voltage Source Inverter'' of Mehdi Ebad and Byeong-Mun Song, Department of Electrical and Computer Engineering, Baylor University, Waco, Texas, USA.

[5]- '' A Novel Current-Tracking Method for Active Filters Based on a Sinusoidal Internal Model'' of Shoji Fukuda, Senior Member, IEEE, and Takehito Yoda.

[6]- '' Rotating Transformation and Resonant Control based Feedback Control Strategy for Dynamic Voltage Restorer System'' of Suxuan Guo, Dichen Liu, Member, IEEE.